UNIVERSITY OF HERTFORDSHIRE

FACULTY OF ENGINEERING AND INFORMATION SCIENCE

7COM0177 COMPUTER SCIENCE M.SC. PROJECT (ONLINE)

FINAL REPORT

JAN 2014

THE IMPACT OF THE CLOUD COMPUTING ON THE SOFTWARE ENGINEERING PROCESS

LUCIENNE BONNICI

STUDENT ID: 09282335

ABSTRACT

It is being forecasted that the renting of software, infrastructure and platform as a service will increase in the coming years. The idea of sharing processing power and resources within a centralized framework seems very enticing. However, to date, cloud computing still has obstacles to overcome especially security and privacy standards.

The aim of this report is to evaluate whether cloud computing will have an impact on the software engineering process and on the quality of software. A CRM web application has been developed and deployed on a public cloud - AppHarbor. This project included various Web 2.0 features such as links to Facebook, Twitter, messaging system, Google Drive, Google Mail and Google Maps integration. Third party software has been integrated within the CRM web application to measure availability, reusability, scalability and performance. This project proposes as a software lifecycle model the SCRUM agile model together with an extension to incorporate the cloud provider role.

The findings led to the conclusion that for small-scale applications hosting on the cloud is ideal and cost effective. During the testing phase the system was unstable for three consecutive days. This downtime had no drastic impact on this project due to its small size. However, to date, clouds are still struggling to successfully adapt to the needs of specific workloads. This report suggests that cloud providers should improve their security measures to guarantee a good quality service.

ACKNOWLEDGEMENT

The implementation of this project and the writing of this report have been the most academically challenging task so far. Without the support and patience of the following people, this project would not have been completed.

I would like to express my deepest gratitude to my supervisor, Dr. Thiago Matos Pinto, for his excellent guidance, patience and prompt advice.

Uncle George, who patiently learned about cloud computing, to be able to proofread this report.

Ruth, my sister, and my parents, for their thoughtful criticism and encouragement.

All my friends, who encouraged me and offered assistance.

Finally, I wish to thank Aaron, my boyfriend, who has always supported, encouraged and believed in me.

CONTENTS:

LIST OF FIGURES:

LIST OF TABLES:

CHAPTER 1 - INTRODUCTION:

Cloud computing is the industry's new buzzword. The idea behind cloud computing is that software and data will be stored on internet servers (clouds) and accessed in real time instead of storing locally on the computer. 12 years ago in 9th January 2001 Apple was the first company to venture into cloud computing and release a service. This was iTunes, a virtual music store. In 2011, Apple has extended this technology to iCloud used by 320 million users. The purpose of iCloud is to sync iOS applications of any iOS-based device.

An infographic presented by Thorhauge S. (2012) shows that 30% of the global population is online. An average user spends 22% of the time on social networks and 19% of the time on emails and communication. Businesses are also investing in cloud services; one of the most popular global cloud computing company known for content management systems, sales and marketing is SalesForce. This company was founded in 1999 and employs 9500 people.

The main reason why companies are choosing the cloud over traditional software projects is because of reusability, computing resources, pay-per-use opportunities and infinite storage resources. Cloud computing eliminates the problem of unnecessary space and power consumption as opposed to typical web hosting, where the user is allocated a fixed server. Another benefit of cloud computing over web hosting is that when the application receives high traffic, the load is automatically distributed across multiple servers.

Since there are many advantages offered by cloud computing, Gartner Inc. (2013) forecasts the global spending on public cloud services will most likely grow from $76.9B in 2010 to $210B in 2016 as shown in *Figure 1*.

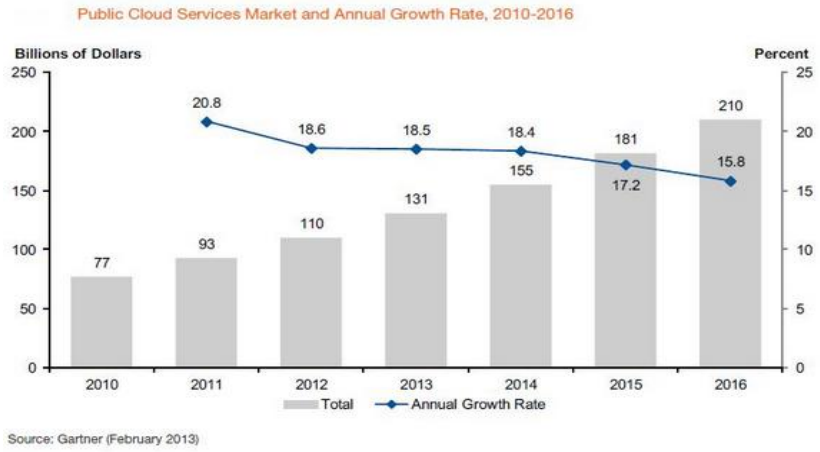


FIGURE 1 - AN ANALYSIS OF THE PUBLIC CLOUD SERVICES MARKET SIZE AND ANNUAL GROWTH RATE BETWEEN 2010-2016 (GARTNER 2013)

Likewise, according to the cloud computing economic report entitled *Economic Potential of Developing Cloud Computing Industry in Northern Ireland* predicts an increase of 18.9% annually between 2011 and 2015 in the cloud computing service market. By 2020, in Northern Ireland, jobs related directly or indirectly to cloud computing cluster will increase by 16,200. Additionally, in a study carried out by SandHill Group (2012) it is forecasted that in the next five years jobs will spur to 4720, 000 and $100b to the global economy.

Development of cloud applications that one would not have ever imagined are nowadays being launched; Real-time traffic and road conditions update, home automation systems, finding friends or services in the vicinity and cloud gaming, to name a few. These examples illustrate the opportunities of cloud computing and the move towards a hyperconnected world. However, cloud computing has several obstacles to overcome. Yau et al (2011) explains that businesses are still skeptical to rely on third-party service providers, and are concerned about service reliability and availability.

1.1 A BRIEF OVERVIEW OF THE SOFTWARE APPLICATION

A CRM (Customer Relationship Management) for a diving company will be developed to demonstrate SaaS (Software as a Service) and to address the research questions. The purpose of a CRM is to improve the company's interactions with customers. Hence, the CRM developed will feature two types of logins: Client and Administrators. The Clients will be able to book diving packages, post reviews, download photos and contact the company via email or instant messaging. The Administrators will manage

users, reviews, bookings and reply to instant messaging. This application will integrate web 2.0 features such as downloading photos from Google Drive, link to Facebook and Twitter and instant messaging. This application will be developed as an ASP.net MVC 3 and deployed on a public cloud – Appharbor. To ensure that software developed for the cloud will improve run-time qualities, statistics will be collected through New Relic, Google Analytics and Code Viewer Metrics (Visual Studio) and these will be analysed.

1.2 Report Outline

- Chapter 1 gives an introduction to the report and an overview of the software project
- Chapter 2 describes the aims, objectives and research questions
- Chapter 3 presents a detailed review of several authors in relation to the purpose of this report
- Chapter 4 gives a justification of the choice of; the type of cloud, cloud provider, software engineering paradigm and statistics tools such as Google Analytics, New Relic and Code Viewer Metrics to develop quality software for the cloud
- Chapter 5 includes data collected between 20th August 2013 and 30th November by the statistical tools mentioned in Chapter 4. These tools collect run-time quality statistics, such as throughput, error rate, uptime, memory and database usage over a period of time.
- Chapter 6 includes an analytic and a critical argumentation of the results obtained in the previous section with reference to the theory mentioned in the literature review.
- Chapter 7 presents the concluding remarks, project evaluation, limitations and further work.

CHAPTER 2 - AIMS AND OBJECTIVES

The aim of this report is to evaluate whether cloud computing will have an impact on the software engineering process and the quality of software. A CRM application for a diving company will be developed and deployed to a public cloud as a service. The development of this project will follow an agile methodology and this should result in a conclusion as to whether software developed for the cloud will improve run-time qualities such as reusability, reliability, scalability, availability and efficiency of the application.

2.1 CORE OBJECTIVES:

The main objectives of this report are:

- To develop an effective, efficient and reliable CRM project by investigating internet-based services, such as Cloud computing;
- To ensure that software developed for the cloud will improve run-time qualities such as reusability, reliability, scalability, availability and efficiency of the software;
- To identify the best software engineering paradigms in relation to cloud computing;
- To ensure that the cost will decrease when software is deployed on the cloud;
- To re-engineer the agile process models by incorporating cloud providers roles.

2.2 ADVANCED OBJECTIVES:

- To provide real-time monitoring function to administrators of the CRM;
- To identify cloud computing implications for software development;
- To analyse the infrastructure security of software hosted on the cloud;
- To identify compliance risks for software developed on the cloud;
- To integrate web 2.0 features such as Google Drive, Facebook and instant messaging in the CRM project.

2.3 RESEARCH QUESTIONS

This report will concentrate on the following questions:

- How will cloud computing impact the software engineering (SE) process to develop quality software based on reusability, reliability, scalability, availability, efficiency, and security attributes?
- Does software in the cloud (Saas) present more benefits than traditional software?
- How will cloud providers fit within the software lifecycle? Will this affect the duration / resources of the project?
- How will cloud computing improve coordination and communication between the team?

CHAPTER 3 - LITERATURE REVIEW:

Facebook and YouTube claim more than 1 billion users per month; Twitter has 500 million accounts, Rackspace hosts over 400,000 websites across 112 countries and an infinite number of Google applications are created and managed daily. A survey published by *PEW Internet & American Life* shows that the highest proportion of cloud computing usage is for webmail and photo storage purposes. Many respondents find online applications very convenient as they can store and access their personal information from anywhere in the world on any device. Statistics show that investment in cloud computing will continue to expand drastically. The International Data Firm (IDC) forecasts that by 2014 $6.4 billion will be spent on the purchasing of server hardware.

This literature review will cover the technical aspects of cloud computing, infrastructures, agile development. It will investigate the software engineering factors vis-à-vis cloud computing, quality, scalability, performance, availability, and costs, and will also present the challenges of cloud computing, including security and environmental issues. The concluding chapter of this literature will summarise the threats and opportunities of cloud computing and traditional software.

3.1 CLOUD COMPUTING – OVERVIEW

Although a universal definition for cloud computing has not yet been formulated, cloud computing may be described as a classification of service frameworks, Software-as-a-Service (SaaS), Platform as a Service (PaaS) and Infrastructure-as-a-service (IaaS).

- Infrastructure as a Service (IaaS): When hardware such as processors, storage and network are delivered to the user.
- Platform as a Service (PaaS): When programming platforms and tools are delivered as a service.
- Software as a Service (Saas). When software applications are delivered as a service.

Figure 2 lists a few examples of the three main levels. The most common free software applications hosted on the clouds (SaaS) that are being used daily are Gmail, Facebook, Dropbox and Google Apps.

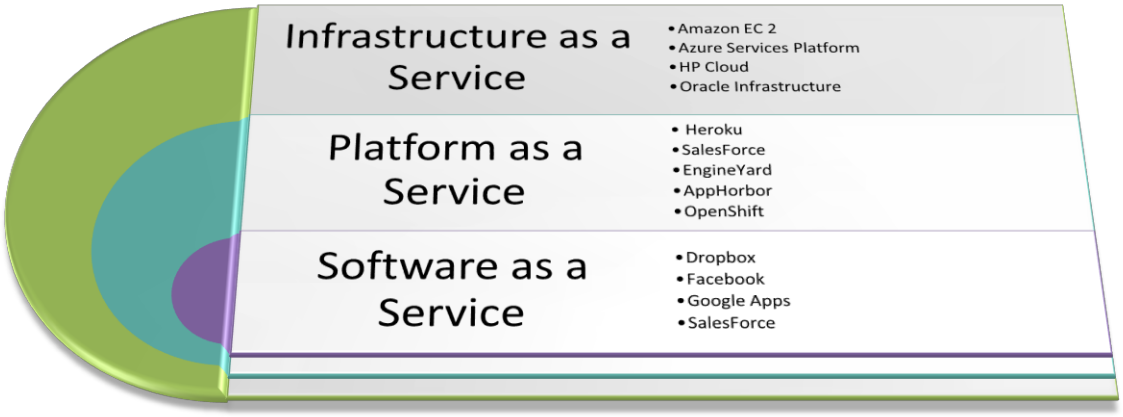


FIGURE 2: THE 3 MAIN LEVELS OF CLOUD COMPUTING. INFRASTRUCTURE AS A SERVICE (IAAS), PLATFORM AS SERVICE (PAAS) AND SOFTWARE AS A SERVICE (SAAS) ARE THE MAIN CLASSIFICATION OF CLOUD COMPUTING. THIS FIGURE IDENTIFIES A FEW POPULAR EXAMPLES FOR EVERY CLASSIFICATION

These services may be deployed on four types of cloud storage: Public, Private, Community and Hybrid. In a public cloud, services are provided over a public network, whereas in a private cloud services are rendered internally on a private network; in a community cloud anyone can contribute and the infrastructure is shared amongst several communities. Hybrid cloud may be a composition of any two or more clouds (public, private or community)

In an article entitled *Comparing public-cloud provider* it is stated that public clouds are more popular than private clouds, albeit the latter are more secure; this will be discussed in the Section 3.6 Challenges.

Table 1 lists several cloud applications together with the programming languages that can be deployed. One can observe that the majority of the cloud applications cater for Ruby on Rails. The main reason behind this is that this general-purpose programming language and framework places emphasis on agile development. However, in this report the application deployed on the cloud will be .Net. The reason behind this decision is that .Net is an object-oriented language, which contributes to agile practices regarding reusability and efficiency and the author is more familiar with the language.

Programming Language	Cloud Provider	Category
Ruby, PHP, Python, .net Framework, Java, Node.js	Amazon Web Services Elastic Beanstalk	PaaS
Ruby on Rails, Java, Node.js. Python, PHP, Scala	AppFog	PaaS
.Net Framework	AppHarbor	PaaS
Java, Ruby, Node.js, Scala	Cloud Foundry	PaaS
.Net Framework, PHP, JavaScript, HTML, Silverlight and CSS.	CodeRun	PaaS
Ruby on Rails & PHP	Engine Yard Cloud	PaaS
Apex Programming language	Force.Com	PaaS
Python, Java, Go, PHP	Google App Engine	PaaS
Ruby on Rails, Java, Node.js, Python	Heroku	PaaS

Java	IBM Smart Cloud	SaaS, IaaS, PaaS
Java, PHP	Jelastic	PaaS
Node.js	Nodejitsu	PaaS
Ruby, PHP, Python, Perl, Java, Node.js	OpenShift	PaaS
Ruby on Rails	Shelly Cloud	PaaS
.NET, Ruby, Java, Node.js and PHP	Uhuru	PaaS
ASP.Net, PHP, Node.js	Windows Azure	PaaS & IaaS

TABLE 1: CLOUD PROVIDER AND PROGRAMMING LANGUAGE. A LIST OF THE POPULAR CLOUD PROVIDERS AND THE PROGRAMMING LANGUAGES THEY CATER FOR. BESIDES PROVIDING A PLAFORM-AS-A-SERVICE (PAAS), TWO OF THE CLOUD PROVIDERS ALSO PROVIDE INFRASTRUCTURE-AS-A-SERVICE (IAAS).

3.2 Cloud computing, Grid computing and Web 2.0

Sometimes the terms 'cloud computing' and 'grid computing' are used interchangeably but their meaning is different. The latter stands for a group of networked, loosely-coupled machines that interact to perform tasks, whereas cloud computing focuses on resource sharing. Zhang et. al (2010) describe grid computing as a distributed computing system that provides unlimited power where anyone can cooperate and access each other's information.

Foster I. et. Al (2008) depict the relationship between Clouds, Grids and other domains, in *Figure 3*. Supercomputers and Clusters are more focused on non-service applications whereas Web 2.0 and Clouds are service oriented. Grid overlaps all domains as it provides a distributed computing infrastructure that spans across multiple virtual organisations.

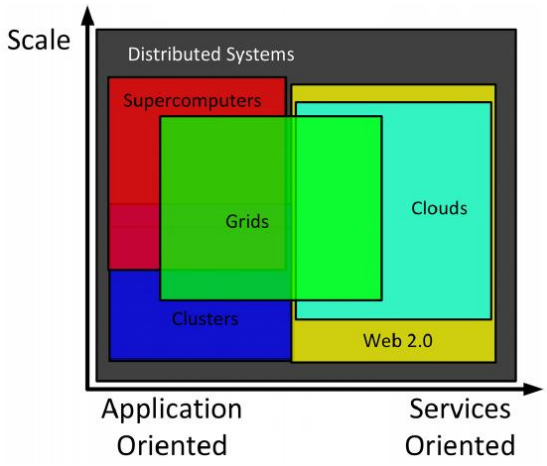


FIGURE 3: 'GRIDS AND CLOUDS OVERVIEW', (FOSTER, I. ET. AL. 2008). GRIDS ARE POPULAR IN NON-SERVICE APPLICATION AND WEB 2.0 APPLICATIONS TO PROVIDE A DISTRIBUTED INFRASTRUCTURE ACROSS MULTIPLE NETWORKS

Figure 3 shows that Web 2.0 shares a large portion of service-oriented applications. The popular GoogleDocs, Gmail, Facebook, Twitter, 4shared, YouTube and LinkedIn, to name a few, are all Web 2.0 applications hosted on the cloud. Furthermore, an article published by VMWare states that cloud computing is based on service oriented architecture of Web 2.0 and virtualisation. Curry R. et. Al (2008) points out that Web 2.0 applications have given access to almost everyone, with minimum or no training required. The availability of social networking, blogs and wikis has given the users many options. However, with all these options it has become very difficult to predict the popularity and the lifespan of an application.

3.3 ADVANCED SOFTWARE ENGINEERING

The aim of Software Development Life Cycle (SDLC) is to enable a smoother transition from one phase to another in order to complete a software project within a timeframe and a budget.

In 1900s agile development was invented and the idea behind agile models is to have short development cycles based on customer feedback. As a result agile development encourages coordination and collaboration; customers are more satisfied with their project because they are continually involved. When agile development is compared to the traditional development, projects are completed in a more realistic timeframe and costs have been reduced because the software developers are still in time to implement last-minute changes. In comparison, in traditional development, the development phase is implemented and it is difficult and more costly to go back and make modifications.

One of the research questions proposed in this report is to analyse whether cloud providers should fit in a software lifecycle and if so, how this would affect the duration and resources of the project. According to SalesForce.com, agile processes model works better on cloud computing platform; software is uploaded on the server and users may provide feedback instantly. Given that all software and data are held in the cloud, there will be serious repercussions if the cloud fails. As mentioned in the sections 3.7 and 3.8, although the probability of failure is not high, the risk remains that the project is stored over a cloud.

In Cloud Computing, when a project is being planned, the cloud provider will be part of the process, as illustrated in *Figure 4*. The main reason is that in the feasibility study one has to determine which is the ideal storage (public, private or hybrid) in relation to size of the application, budget and security measures; for example, a bank application should be more secure than a simple DBMS for a small company. According to Kherajani M. et Al. (2012) the role of the cloud providers will be to determine how many software developers are needed, the cost and duration estimation of the project, component reuse, risk and configuration management and quality assurance.

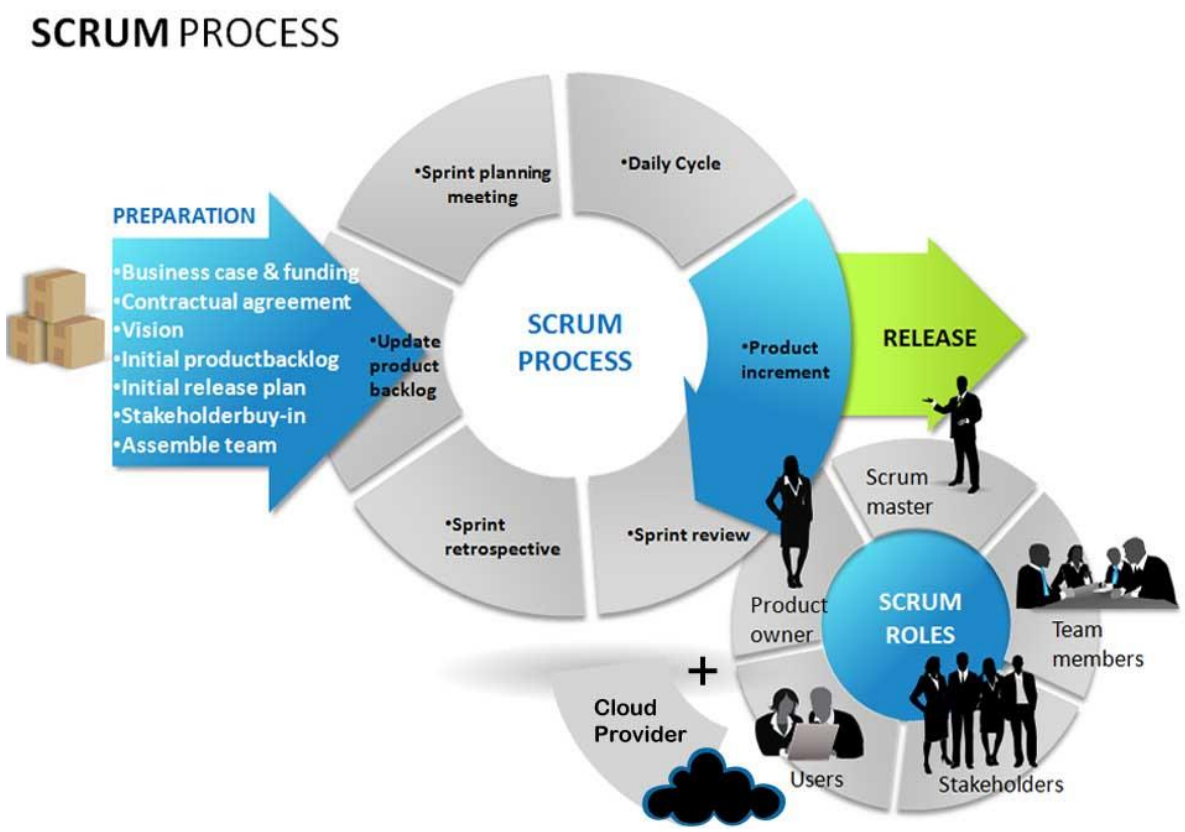


FIGURE 4: SCRUM PROCESS (MAYNARD C. 2012 AND AMENDED BY AUTHOR). AN OVERVIEW OF AN AGILE METHODOLGY - SCRUM TOGETHER WITH THE TEAM MEMBERS INVOLVED INCLUDING THE CLOUD PROVIDER WHO WILL BE INCLUDED IN THE TEAM IN CLOUD COMPUTING APPLICATIONS

When involving more people in the team, in this case, the cloud provider, this may not always be advantageous. It is clear that the more people involved, the more difficult it is to communicate. As a matter of fact, Guha R. & Al-Dabass D. (2010) states that software projects fail because of poor communication and coordination between the team members involved in the project. However, nowadays many companies are promoting social networking, particularly chatting, to overcome communication barriers. This solution might not be one-size-fits-all but it is very useful when companies are based across the globe.

Projects that are already implemented can be migrated to the cloud. Although this might not sound straightforward, with careful planning migration can be viable. Da Silva E. et. Al. (2012) emphasise that before making the decision to migrate to the cloud, the organisation has to analyse the technical and organisational constraints, and consider its business requirements. To support the cloud computing business model, the authors point out that SaaS models must implement a way to register the use of

services. When developing an IaaS model, the developers should take into consideration load balancing, automatic scalability and data storage services. Research carried out by Esparza-Peidro J. et. Al (2011) concluded that IaaS model is ideal to deploy pre-existing systems and the PaaS is more adequate for new applications.

In his article Draheim D. (2012, p. 2) raises the following questions:

> "How to organise – conceptually and technologically – the tight integration of software engineering tools in the cloud? What are the appropriate cloud-enabled patterns of management and work organization for very large, distributed teams?"

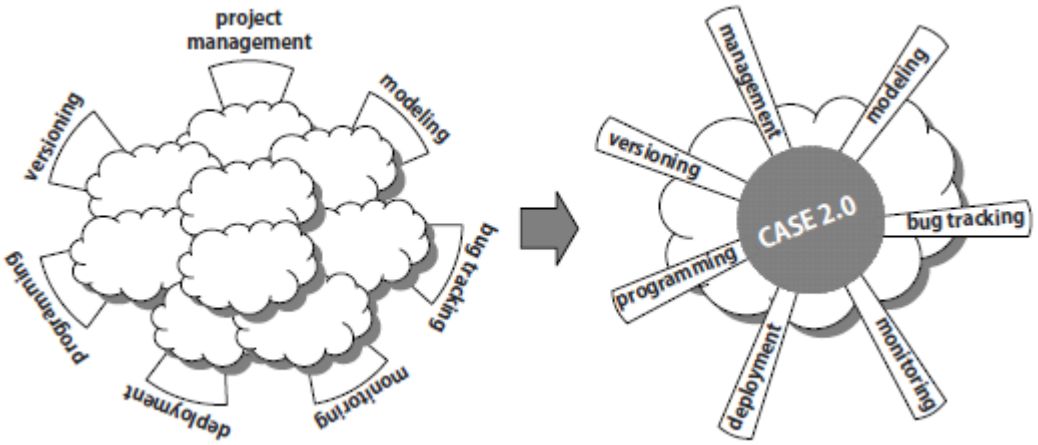


FIGURE 5: 'ALIGNMENT OF CASE TOOL SERVICES IN THE CLOUD' (DRAHEIM D. 2012). CLOUD-AIDED SOFTWARE ENGINEERING (CASE 2.0) WILL AID THE INTEGRATION OF CLOUD COMPUTING TO THE TRADITIONAL SOFTWARE PROCESS

Figure 5 explains that one cannot simply shift the software engineering tools suite to the cloud because they will lead to project failure. Case 2.0 (cloud-aided software engineering) promises to integrate the whole new software revolution.

3.4 QUALITY

It is crucial to maintain a good quality service, software-as-a-service provides many benefits to customers; high availability, pay-per-use, and free maintenance and updates. Boehm et. Al (1978) proposed fifteen quality attributes for conventional software and these attributes relate to software dependability, usability, efficiency and maintainability. Lee Y. et. Al (2009) developed a quality model for evaluating SaaS based on IEEE 1061 framework. This model includes additional quality features to cater for SaaS.

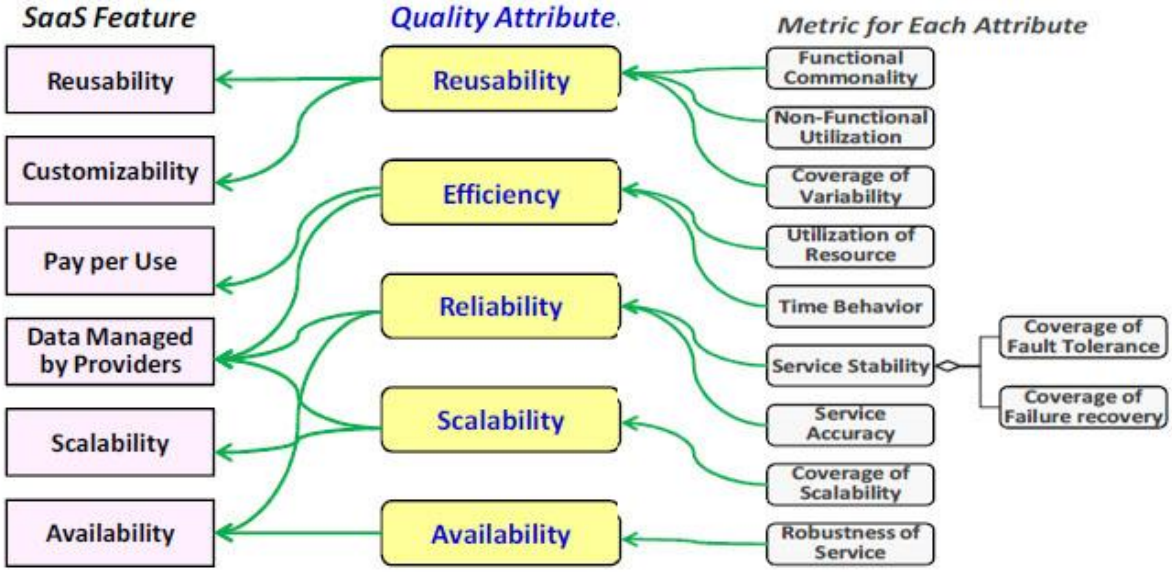


FIGURE 6: 'MAPPING FROM FEATURES TO QUALITY ATTRIBUTES' (LEE Y. ET. AL 2009). A LIST OF THE QUALITY FEATURES OF AN SAAS. THIS FIGURE ALSO INCLUDES THE 5 MAIN QUALITY ATTRIBUTES: RESUABILITY, EFFICIENCY, RELIABILITY, SCALABILITY AND AVAILABILTY TOGETHER WITH THE METRICS FOR EACH ATTRIBUTE.

Figure 6 shows the relationship between the SaaS features and the quality attribute together with the metrics for each quality attribute. ISO/IEC 9126 and Consortium for IT Software Quality (CISQ) in 2012 published a set of software quality attributes mainly relating to Reliability, Performance efficiency, security and maintainability.

To improve quality standards, service providers opt to integrate third-party plugins to monitor and analyse the quality of their applications in real-time. Some of the plugins available may be used to customise metrics, such as StatsMix and Hosted Graphite; other plugins have specific metrics, for example, to measure error rate or response time. Cloud providers integrate their own performance tools; for example Windows Azure utilises DiagnosticMonitorConfiguration class.

3.4.1 REUSABILITY

Lee et. Al (2009), designed a quality model for evaluating SaaS. The authors measure the attribute reusability by using three metrics: Functional commonality, non-functional commonality and coverage of variability.

$$Reusability = W_{FC}.FC + W_{NFC}.NFC + W_{CV}.CV$$

Where W_{FC}, W_{NFC} and W_{CV}, are the weights for each metric, the sum of which is 1.

$$FC = \left(\sum_{i=1}^{n} \frac{number\ of\ requirements\ appling\ ith\ functional\ feature}{total\ number\ of\ requirements\ analyzed\ in\ the\ domain} \right) / n$$

$$NFC = \left(\sum_{i=1}^{m} \frac{\text{number of requirements appling ith non - functional feature}}{\text{total number of requirements analyzed in the domain}} \right) / m$$

$$CV = \frac{\text{(number of variation points realized in the SaaS)}}{\text{(number of variation points in the domain)}}$$

This model calculates the number of clients subscribed and the number of features required by the clients using the services. In this case, Lee et. Al (2009) interprets reusability and its factors as an indicator to measure ROI (return on investment) of service providers. Baklizi, M. and Alghyaline S. (2011) used this model in their research and to obtain the desired results the authors made use of questionnaires and assigned points to the features to be able to measure the amount of subscribers and features required by service consumers.

However, in this report the aim of this attribute (reusability) is not to measure the amount of subscribers and its features provided but to evaluate the use of reusable objects and software in cloud computing.

In an article published by Erni K & Lewerentz C. (1996) they presented the following quality model (*Figure 7*) and the factors and metrics related to Reusability. Chidamber S.R. & Kemerer C.F. (1994) agree that reusability can be measured by using Weight Methods per class (WMC), Depth on inheritance Tree (DIT) and Number of Children (NOC), Coupling between Object Classes (CBO).

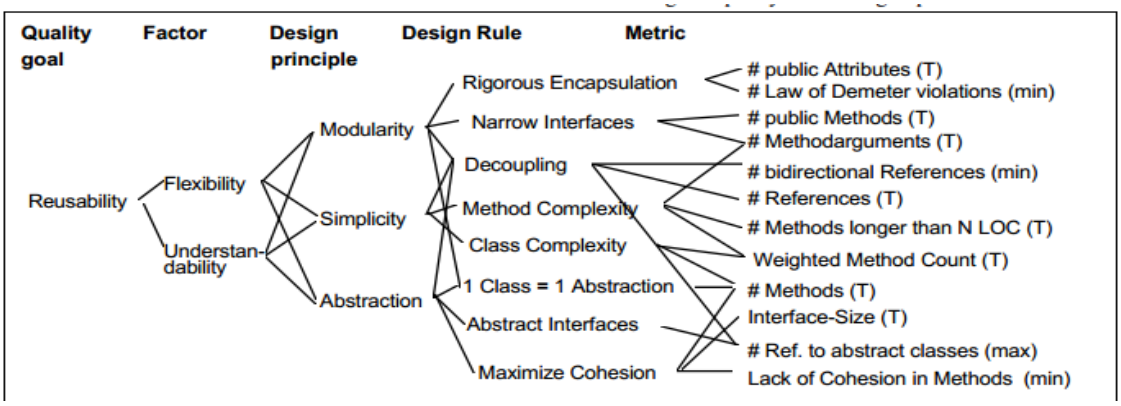


FIGURE 7: 'EXTENDED QUALITY MODEL FOR THE DESIGN OF OBJECT-ORIENTED SOFTWARE' (ERNI, K.; LEWERENTZ, C., 1996). THE MAIN FEATURE OF OBJECT-ORIENTED SOFTWARE IS REUSABILITY. THIS FIGURE IDENTIFIES THE FACTORS EFFECTING RESUABILITY AND A LIST OF METRICS TO MEASURE RESUABILITY

To be able to calculate maintainability index, cyclomatic complexity, class coupling, depth of inheritance and lines of code, Visual Studio 2010 includes an extension called Code Metrics Viewer. This extension integrates with Power Tool 10.0. This version can only be used on Visual Studio 2010, another version is available for Visual Studio 2012.

3.4.2 RELIABILITY

Erl T.; Mahmood F.; Puttini R., (2013) describe a *reliable* SaaS as a service with minimum time between failures and a guaranteed rate of successful responses.

Lee et. Al (2009) measures reliability based on two metrics service stability (in relation to number of faults recorded) and service accuracy:

$$Reliability = W_{CFT}.CFT + W_{CFR}.CFR + W_{SA}.SA$$

Where W_{CFT}, W_{CFR} and W_{SA} are the weights for each metric, the sum of which is 1.

The metrics derived to measure reliability are Coverage of Fault Tolerance (CFT), Coverage of Failure Recovery (CFR) and Service Accuracy (SA) and are measured as follows:

$$CFT = \frac{(number\ of\ faults\ without\ becoming\ failures)}{(total\ number\ of\ faults\ occurred)} \qquad CFR = \frac{(number\ of\ failures\ remedied)}{(total\ number\ of\ failures)}$$

$$SA = \frac{(number\ of\ correct\ responses)}{(total\ number\ of\ requests)}$$

To measure service stability, Lee et. Al (2009) define fault as an abnormal condition that causes system malfunction and failure as the inability to perform a requested function.

As a contingency plan, service providers integrate plugins in their system to monitor system reliability. Popular plugins include Airbrake and AppFail; an exception monitoring service that can be integrated with ASP.net and Javascript. New Relic is another plugin used for monitoring .Net applications and one of its features is to measure the error rate. This plugin will be integrated in this research.

3.4.3 SCALABILITY

Scalability is the ability to handle a growing number of users and resources requesting services without failing. The better the application scalability, the higher it can handle simultaneously. Gao J. et Al (2001) states that providing elastic scalability for SaaS in a cloud proves to be a main benefit and an important feature in cloud computing. Guha R. & Al-Dabass D. (2010) and Cunsolo, V.D. et. Al (2010) agree that due to the adoption of the virtualisation technique in cloud computing, resources are managed more efficiently because these can be shared amongst all users accessing the cloud. This would provide a higher degree of resource utilisation and on-demand scalability.

Wu J. et. Al. (2009) present a model to improve the scalability of software as a service (SaaS). The analysis of this model is based on the volume of requests and the availability of servers. Chieu, T.C. et Al. (2011) describe the main four factors that measure the scalability and the performance of a web application in cloud computing. The four factors are: the number of users using the application simultaneously, the number of active connections, number of requests per second and average response time per request.

The universal Scalability Law was introduced in 1993 by Gunther N.J. and this presents a universal function to measure scalability:

$$C(N) = \frac{N}{1 + \sigma(N-1) + \kappa N(N-1)}$$

The parameters α (alpha) represents the queuing for shared resources (throughput) and β (beta) represents the waiting time for data to become responsive (response time). If β is 0, the Universal Scalability Law can be simplified to Amdahl's Law:

$$C(N) = \frac{N}{1 + \sigma(N-1)}$$

3.4.4 AVAILABILITY

Internet service providers are obliged to provide a service level agreement (SLA) to their customers to guarantee a quality of service. Similarly, an SLA is also drawn up by cloud providers highlighting one of the most important characteristics - service availability.

Avizienis A. (2004) defines availability as "*the readiness for correct service*", in other words, the contingency of providing a service according to defined requirements.

Cloud computing and SaaS have become increasingly popular during these last 5 years. In spite of this, Brandon J. (2013) states that cloud service providers are still struggling to guarantee service reliability. In an interview, Gavin Walker, CIO of National Air Traffic Control Services, said that not even Microsoft was able to provide sufficient guarantee for security and availability. In my opinion, software companies were (and still are!) busy migrating their projects to the cloud rather than focusing on the QoS (quality of service). In their work, Undheim A. et. Al(2011) represent QoS as dependability, performance and security. The authors illustrate an SLA as follows:

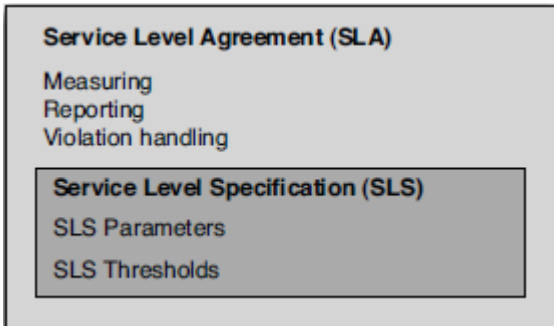


FIGURE 8: 'THE STRUCTURE OF AN SLA' (UNDHEIM A. ET. AL 2011). AN SLA IS AN AGREEMENT BETWEEN THE CLOUD PROVIDER AND THE CLIENT. THIS SHOULD ALSO INCLUDE A QUALITY OF SERVICE (QOS) AGREEMENT IN THE SLS.

The cloud framework needs to adjust to changes in user requirements, network infrastructures, unforeseen circumstances and environment issues. To avoid paying fines, cloud providers need to ensure that when the need arises they react accordingly. Undheim A. et. Al (2011) proposes allocating more resources and, when the current machine is overloaded, move to another physical machine.

According to Wazzan M. & Fayoumi A. (2012) users will choose a cloud provider that ideally provides five nines (99.999s%) availability and 24/7 support. When designing a cloud service model a service protection technique must be incorporated. There are 3 service protection techniques 1:1, 1:N or M:N. The authors proposed a model to evaluate availability in a hybrid cloud computing architecture using a 1:1 service protection. As a result, the availability of the whole system increased after adding the public service protection.

Qian H. et. Al (2011) , Avizienis A. et Al. (2004) and Dai Y.S. et. Al (2006) state that in a data centre failures range from software related, configuration, overflow and timeout failure, human, networking related errors, hardware errors and database failures. These failures are described in more detail in Section 3.7. Nonetheless, Avizienis A. et Al (2004) explain that in cloud computing, companies choose inexpensive and off-the-shelf hardware and rely on software to handle failover. This has been regarded as less expensive than investing in robust hardware, which will eventually still fail due to the large size of cloud data centres. In their study Qian H. et. Al (2011) proposed a model that can help online service providers to acquire better performance from cloud service providers, such as higher reliability and more bandwidth availability while maintaining quality of experience (QoE). To be able to monitor and improve service availability, Oracle (2012) defines the following formula:

Availability = Average Time to Failure (ATTF) / (average time to failure (ATTF) + average time to recover (ATTR))

VMware vFabric™ Hyperic® monitors operating systems, middleware and applications running in physical, virtual and cloud environments. This tool collects metrics that reflect availability, performance, utilisation and throughput. VMware vFabric™ tracks and records events, and enable the administrator to manage resources, as shown in *Figure 9*.

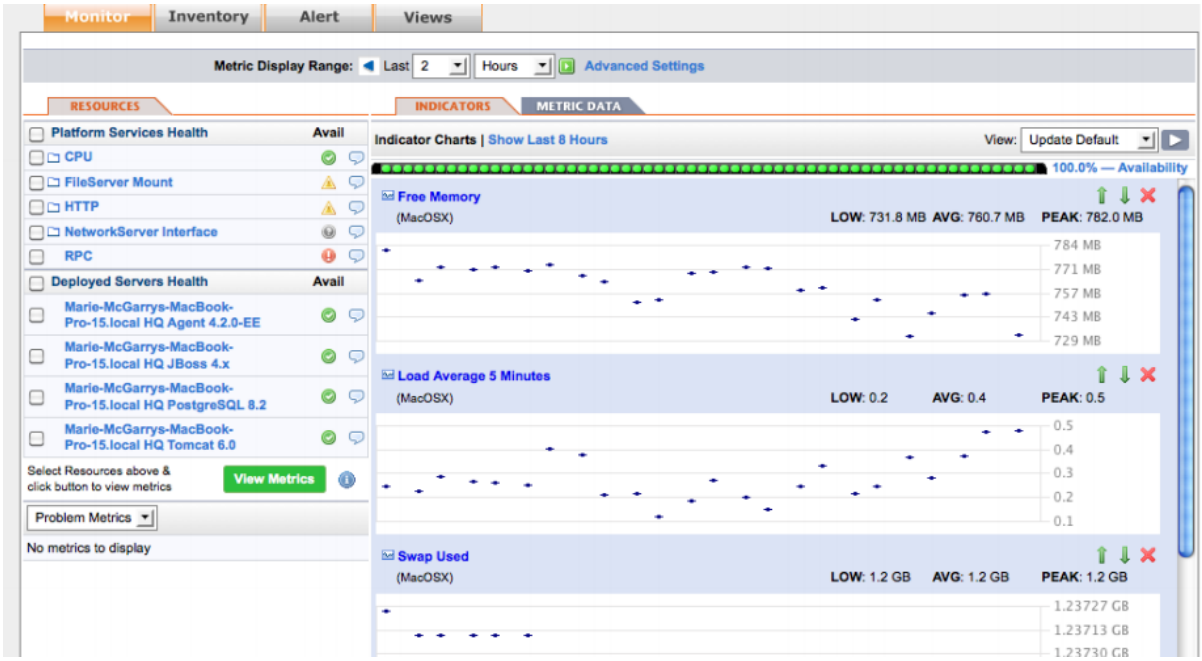


FIGURE 9: VMWARE VFABRIC™ HYPERIC® DASHBOARD. THIS IS A SAMPLE OF SOME OF THE METRICS MONITORED SUCH AS AVAILABLE MEMORY AND AVERAGE LOADING TIME OF A SYSTEM.

3.4.5 PERFORMANCE (EFFICIENCY)

3.4.5.1 PERFORMANCE METRICS

There are various metrics and techniques on how to assess performance, as the term *performance* is debatable.

Chiew K. T. (2007) refers to performance as how efficiently the system uses resources such as CPU, memory of disk storage. Similarly, the main parameters outlined by Alhamad et al. (2010) are CPU performance, storage disks and network infrastructure. Lee et. Al (2009) describe this attribute as a measure of how fast the system responds to a request and propose a formula based on these two metrics: utilisation of resources and time behaviour.

$$Efficiency = W_{RU}.RU + W_{TB}.TB$$

Where W_{RU} and W_{TB} are the weights for each metric, the sum of which is 1.

The computation for Resource Utilisation (RU) and Time Behaviour (TB) are measured as follows: $RU = \frac{(amount\ of\ allocated\ resources)}{(amount\ of\ pre\text{-}defined\ resources)}$ $TB = \frac{(execution\ time)}{(total\ service\ invocation\ time)}$

In this report the term performance refers to the speed; response time of downloading the web application, in relation to the cloud vendor. A large amount of third party plugins are available to measure performance. Some of the plugins are free whereas others are against payment. A free and easy to use application is www.alertra.com, the aim is to monitor servers around the globe. The developer simply provides the URL and this application generates the Server response, Check time, Request time, Content length, Data Size, Transfer Time and Transfer rate from various different cities in Asia, North America, Europe and U.S.A..

Internet service providers may also affect the performance of the system, since cloud applications rely on internet providers. A free website called www.speedtest.net may be used to measure the consistency of the speed from the internet provider. A sample of results is illustrated in *Figure 10*:

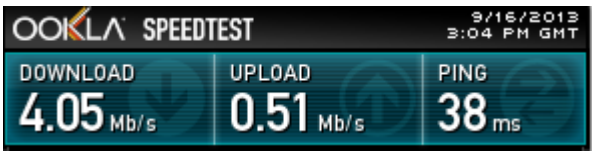


FIGURE 10: INTERNET PROVIDER SPEED TEST. THE THIS TEST WAS GENERATED THROUGH WWW.SPEEDTEST.NET TO ANALYSE THE INTERNET SPEED OF THE CURRENT LOCATION. IT SHOWS THAT DOWNLOAD SPEED IS OF 4.05MB PER SECOND AND THE UPLOAD SPEED IS OF 0.51MB PER SECOND.

Another popular and free application is GoogleAnalytics. This application is able to measure the average time in seconds for page load, redirection, domain lookup, server connection time, server response time and page download time.

3.4.5.2 Performance Benchmarking

According to Kotsis (2006), performance benchmarking is a measuring process used to compare more than one system. A study carried out by Wang et al. (2010) suggests AOSC (automatic optimisation schema for cloud storage). This schema makes use of partitioning of data and replication and migrating the data to an ideal site to enhance performance. Amazon Elastic Cloud Computing (EC2) data centre carried out a few experiments by using AOSC and results showed higher stability and performance in the cloud. Also, Zheng L. et. Al (2013) proposed the concept of *Boosting Metric,* to present a holistic performance of a cloud service to the customers.

In their study, Jackson K.R. et. Al (2010) show that benchmark suites are already being used; HPCC is a popular benchmark suite to identify high performance computing capabilities of cloud services. Using Metrics to evaluate services are important not only for cost-based analysis but also for cloud provider selection.

3.5 COST

Most of the cloud services have a pay-per-use scheme; the client will only need to provide the credit card number at the start of the process. This scheme has enticed many organisations such as government, schools and small enterprises to shift their applications to the cloud. Very few provide a free cloud service; some of them are free for 15 days or 1 month. CodeRun and SalesForce only offer per-month pricing plans.

Table 2 highlights the pricing plans for several cloud providers:

Cloud Provider	Pricing Plan
Heroku	512MB RAM, 1x CPU Share, $0.05/hour
Amazon (EC 2)	1.3 GB Ram, 160 GB storage, $0.15/hour
Engine Yard Cloud	(First 500 hours free) 1.7GB RAM, 160 GB storage, $0.05/hour
Go Grid	2 GB RAM, 100 GB storage, $0.16/hour
RackSpace	2GB RAM, 80GB storage, $0.22/hour
OpenShift	(512MB, 1GB storage, free) 1GB RAM, 6 GB storage, $0.10/hour
Google App Engine	500MB storage, free
HP cloud	2GB RAM, 60GB storage, $0.12/hour
Appharbor	1 worker Free, 2 workers $49/month
Windows Azure	(90-day free trial) 1.75GB RAM, 1GB storage, $0.10/hour
Force.Com	(30-day trial), 20MB storage, $10 /app/user/month
CodeRun	(256MB Ram, 5GB Storage, 14-day trial) 512MB Ram, 5GB storage, $49 per month

TABLE 2: PRICING PLANS FOR CLOUD APPLICATIONS. PRICES INCLUDE HOSTING AND STORAGE SPACE

Table 2 shows that prices range from $0.05/hour to $0.22/hour, depending on storage space and memory. Whereas, the billing of a standard dedicated server is normally higher than $80 per month. For example, GoDaddy pricing plan starts from $117 per month, Ipage starts from $149 per month and the standard price for JustHost is $152.

In a research carried out by Roloff E. et. Al (2012), the performance and price of three cloud providers (EC2, Azure, RackSpace) were compared using the following two metrics:

i. Cost Efficiency – a value which represents the cost per hour of running equal work on various systems.

*Cost efficiency = Average Performance * Cost per hour*

ii. Breakeven point – represents the number of days per year, after which execution on a cluster becomes cheaper than on the cloud.

Break Even Point = Yearly Cost / (24 X Cost Efficiency)

Roloff E. et. Al (2012), found that by comparing a traditional cluster and three cloud providers; Amazon Elastic Compute Cloud, Rackspace and Microsoft Windows Azure, the latter is estimated to be 41% more efficient, thus, it costs 41% less to perform the same task in the cloud than in a cluster. Li A. et al. (2011) measure performance of different cloud solution vis-à-vis computing, network, database and storage of the four most popular cloud providers: Amazon EC2, Windows Azure, Google AppEngine and Rackspace. They conclude that no ideal cloud exists in terms of performance, but the features of every cloud are designed for a specific area.

In a report called *Economics of Cloud Computing* the authors use three metrics to analyse the economic benefit of public, private and hybrid cloud. The three metrics are *net present value (*calculates each cloud discounted net benefits less the cloud's discounted one-time investment cost*), benefit-to-cost ratios (*calculates each cloud's discounted net benefit divided by its discounted investment costs*) and discounted payback period (*the number of years it takes for each cloud to break-even total investment costs with accumulated annual benefits). As a result, the total life-cycle costs, over 13 years, of a private cloud are 49% more than that of the public cloud. The constituent to this increase is the $7 million investment phase. This includes the

technical and labour planning and the purchasing of servers and hardware. It is also noted that by the 3rd year the public cloud will start expanding its profits, whereas the private cloud will have a profitable turnout a year later.

Costs/Economic Metrics	Status Quo: 1,000 Server (Non-Virtualized) Environment	Scenario 1: Public Cloud	Scenario 2: Hybrid Cloud	Scenario 3: Private Cloud
Investment Phase Cost	$0	$3.0	$6.1	$7.0
Operations & Support Phase Costs	$77.3	$22.5	$28.9	$31.1
Total Life-cycle costs	$77.3	$25.5	$35.0	$38.1
Economic Metrics:				
Net Present Value	N/A	$41.8	$33.7	$31.1
Benefit-to-Cost Value	N/A	15.4	6.8	5.7
Discounted pay-back period (years)	N/A	2.7	3.5	3.7

TABLE 3: 'ECONOMICS OF CLOUD COMPUTING' (ALFORD T. & MORTON G. 2010). THIS TABLE SHOWS THAT THE PUBLIC CLOUD RETURNS THE HIGHEST BENEFIT-TO-COST VALUE AND NET PRESENT VALUE.

3.6 CHALLENGES

The preceding sections depicted cloud computing as an all-powerful technology but cloud computing is still struggling with a myraid of threats, the topmost being privacy and security. A report propounded by *Cloud Security Alliance* states that Google reported the highest number of attacks, followed by Amazon and Microsoft. Presumably these three providers ranked highest because they are also the most popular clouds.

Behl A. (2011) mentions four perils of cloud computing: hacking, data loss, service disruption and loss of control. Clouds are vulnerable since they are more associable than private networks and have more interfaces. Data loss is another threat of cloud computing; whilst migrating data to the cloud, data may be lost. Likewise, migration may also lead to service disruption and this may lead to customers complaining. Another concern is loss of control. When organisations migrate to the cloud, providers host their data or services anywhere in the cloud; organisations are thus not in control of vital data or familiar with the security mechanisms.

In an article entitled *Caught in the Clouds: The Web 2.0, cloud computing and Privacy?,* the authors show that the 2002 ePrivacy Directive is not clear and open to interpretation. Cookies nowadays can track user activities in real time without the user being aware by storing the users' IP and the processing of unique identifiers.

Data stored in the cloud may be located in any part of the globe. This may present privacy threats. Data privacy laws vary from one country to another. For example, under EU law, personal data can only be gathered at the customers' discretion and if the customer feels that personal data was misused in EU s/he has the right to complain. On the other hand, in the Information Technology Act 2008 of India there is no definition of 'personal data', 'processing' or 'consent' and does not meet any international standards. Thus, not having a common law across the world may make personal data vulnerable. Many cloud providers are opting to comply with the EU law and ensure that data is segregated in the EU clouds.

Sabahi F. (2011) and Behl A. (2011) propose a few solutions to the above threats; improve authentication requests, control administrators and users' properties and rights, set parameters to control access to network services, operating systems and applications.

3.7 FAILURES

The infographic published by IEEE Spectrum illustrates that the highest number of faults were reported in 2008, presumably due to the influx of cloud applications. From 11,491 incidents reported over five years, between Jan 2008 and Feb 2012, the top three were: 29% security issues, 25% data loss and leakage issues and 18% hardware failures.

Qian H. et. Al (2011) and Avizienis A. et Al (2004) agree that the main factors that contribute to failure are: Software Failures, network failures, power failures and server failures. However, the *Cloud Security Alliance* proposed the following failure categories: natural disaster, service closure, cloud-related malware, inadequate infrastructure planning and hardware failure, which was also mentioned by Qian H. et. al (2011) and Avizienis A. et al (2004) as being one of the main factors leading to cloud failures. In 2013, CloudTweaks published an article stating that 70% of the hardware failures.

3.8 SOLUTIONS

As a solution Avizienis A. et Al (2004) deployed three different schemes by using replicas in different physical locations and this affected fault-tolerance and availability. The three schemes are: placing all replicas in the same cluster, placing replicas in two different clusters but in the same data centre, and place replicas in two different data centres. Kavalionak H. & Montresor A. (2012) agree that replica management in cloud-based applications is essential to balance service reliability and economical costs.

The authors of CloudTweaks suggest the following techniques in the case of hardware failure: to deploy different RAID levels in the memory architecture. RAID levels are improved with decision tree techniques responsible for automatic error detection. Another suggestion is to deploy memory management techniques with efficient hardware devices and the ultimate suggestion is Transaction management system with optimised support for extended memory devices.

Avizienis A. et Al (2004) and Undheim A. et. Al (2011) explain the following network model - the data centre consists of two duplicated layers. Level 1 connects all servers in one cluster and level 2 connects all level 1 switches from every cluster available. Both levels are connected to two WAN networks since the assumption is to be multihomed to two independent internet service providers.

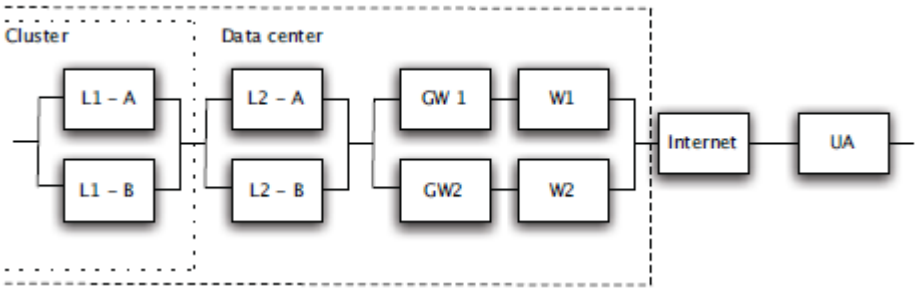


FIGURE 11: 'NETWORK MODEL' (UNDHEIM A. ET. AL 2011). L1 CONNECTS ALL SERVERS IN ONE CLUSTER AND L2 CONNECTS ALL L1 TO THE GATEWAYS AND WIRELESS AREA NETWORKS.

The uninterrupted power supply (UPS) unit will supply power to the data centre and provides backup in case of power failure.

CHAPTER 4 - METHODOLOGY:

In this chapter I will provide a justification for the selection of a public cloud to deploy the application, the choice of the cloud vendor, the selection of third-party software to gather real time statistics; New Relic, Google Analytics and Code Viewer Metrics (Visual Studio) and a justification why an agile software process model has been selected for SaaS.

4.1 CLOUD SELECTION

The application has been deployed on a public cloud. The two main factors that led to this choice were research and costs. Research carried out by Gartner Inc. (2013) showed that the public cloud spending would increase in the coming years. For this reason, evaluating software deployed on a public cloud would be more beneficial. Besides, it was cheaper to deploy on this type of cloud, since this application was developed for research purposes only.

4.2 BUILDING THE APPLICATION AND CLOUD VENDOR SELECTION

After analysing the pricing schemes of the popular public .Net Cloud platforms listed in Table 4 - Public cloud and pricing, AppHarbor was selected to deploy my application and gather results.

Public .net Cloud Platforms	Price
Amazon Web Services (AWS)	12 months free trial (must provide credit card details). $0.15/hour
AppHarbor	Free for a simple plan (no credit card required)
Code Run	14-day trial. $49 per month
Uhuru	Free (Beta Version)
Windows Azure	90day free trial (must provide credit card details). $0.10/hour

TABLE 4 - PUBLIC CLOUD AND PRICING. LISTS THE PRICING PLANS AND TRIAL PERIODS OF THE PUBLIC .NET PLATFORMS.

AppHarbor was selected as the cloud vendor for this report because it was completely free to deploy an application, while the AWS, Code Run and Windows Azure promoted only a trial period free of charge. Uhuru was free but it was still in a

beta version during the selection process. Code Run offered a free 14-day trial period and had very limited documentation online.

4.3 Technical Details of the application developed

A CRM (Customer Relationship Management) for a diving company was developed using ASP.net MVC 3 to demonstrate SaaS (Software as a Service) and to address the research questions. This project was implemented using Visual Studio 2010 and Microsoft SQL Server Compact 4.0. The application was integrated into AppHarbor through Microsoft's open source subversion site – CodePlex and could be accessed through http://luciennecrm.apphb.com

The diagram, as drawn in *Figure 12*, shows how the source code developed on the local host was transferred to the public cloud Appharbor through Codeplex.

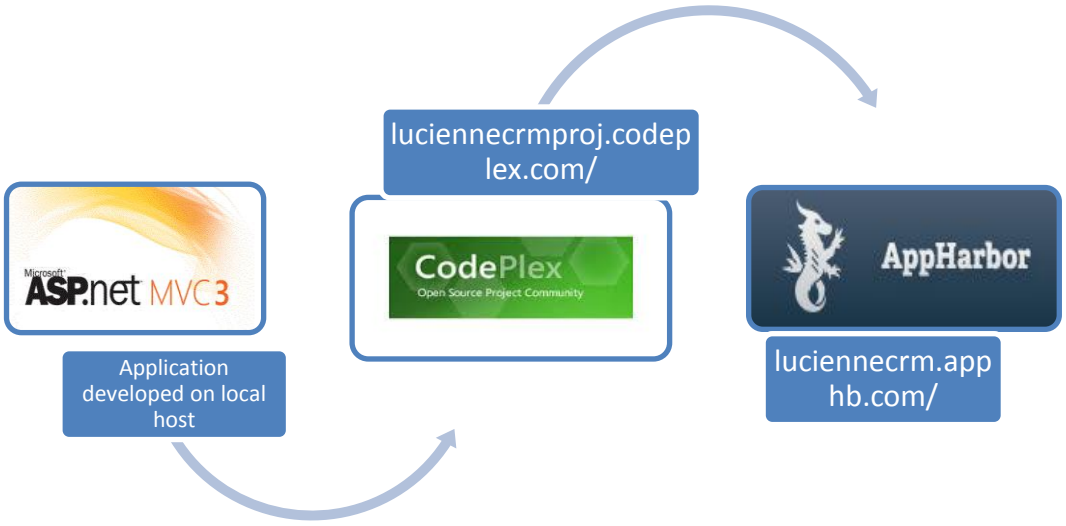
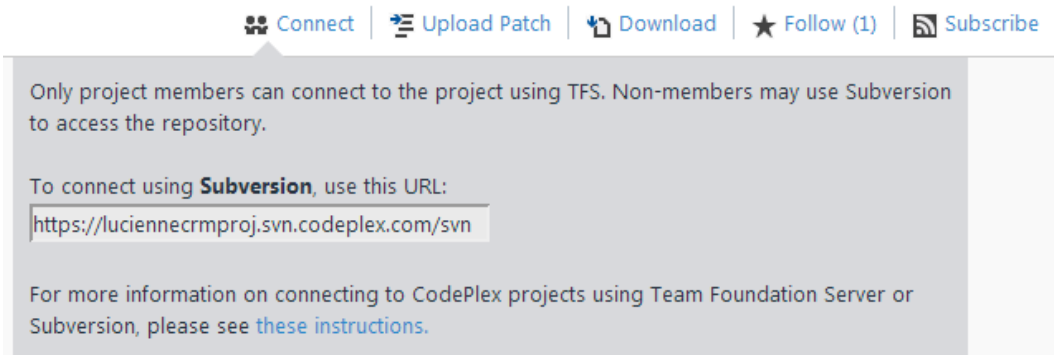


FIGURE 12 – THE FLOW OF DATA FROM THE LOCALHOST TO THE PUBLIC CLOUD. THE APPLICATION IS DEVELOPED ON THE LOCALHOST, UPLOADED TO A VERSION CONTROL CALLED CODEPLEX AND THROUGH CODEPLEX THE APPLCIATION IS DEPLOYED TO THE PUBLIC CLOUD – APPHARBOR

This link below was used to connect to the version control and update the changes:


Connect
Upload Patch
Download
Follow (1)
Subscribe
Only project members can connect to the project using TFS. Non-members may use Subversion to access the repository.
To connect using Subversion, use this URL:
https://luciennecrmproj.svn.codeplex.com/svn
For more information on connecting to CodePlex projects using Team Foundation Server or Subversion, please see these instructions.

Appharbor provides an SQL Server sequelizer for its applications, to ease syncing betweeen intances. In this project the connection string used was:

```
Server=27379589-933a-4441-8f9f-
a1d2014475de.sqlserver.sequelizer.com;

Database=db27379589933a44418f9fa1d2014475de;

User ID=kvpkwrrerebapvio;

Password=CycoSPfXAd4N3BBgWpwKsGmU2RLboq37fuktT2VRTV8XVapoQ5muC
Rq4mqvbceHj;
```

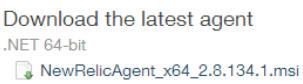
4.4 Statistical tools selected

The statistical tools used in this research to track run-time qualities; Resuability, Reliabilty, Scability, Availability and Efficiency were gathered by third-party Plugins integrated into the system. These plugins were New Relic and Google Analytics.

4.4.1 New Relic

The New Relic plugin was chosen as the performance management tool to monitor the CRM application, because New Relic was integrated with the AppHarbor platform and access was available through a single-signon link of the same interface. Besides, it was the only free tool that could be easily integrated with .Net Application. The purpose of this plugin was to monitor throughput, apdex index, error rate, response time, load time, memory and database in real-time.

To set up New Relic, the plugin was downloaded from AppHarbor and the license key `d508517bc17df7c5171a95d03b6b2e23bad7fc88` was used for installation. The latest X64bit agent was downloaded from the New Relic Site and license key was used during registration:



After installation, New Relic started tracking the application deployed on AppHarbor and generated a weekly report. In case of downtime, the user was alerted *via* email and the downtime period was described as shown in

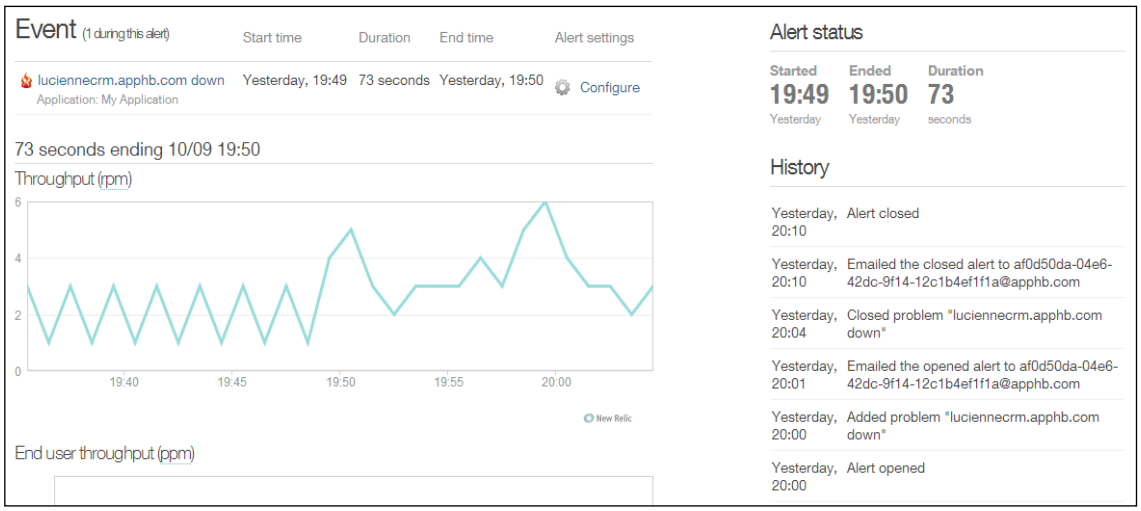


Figure 13:

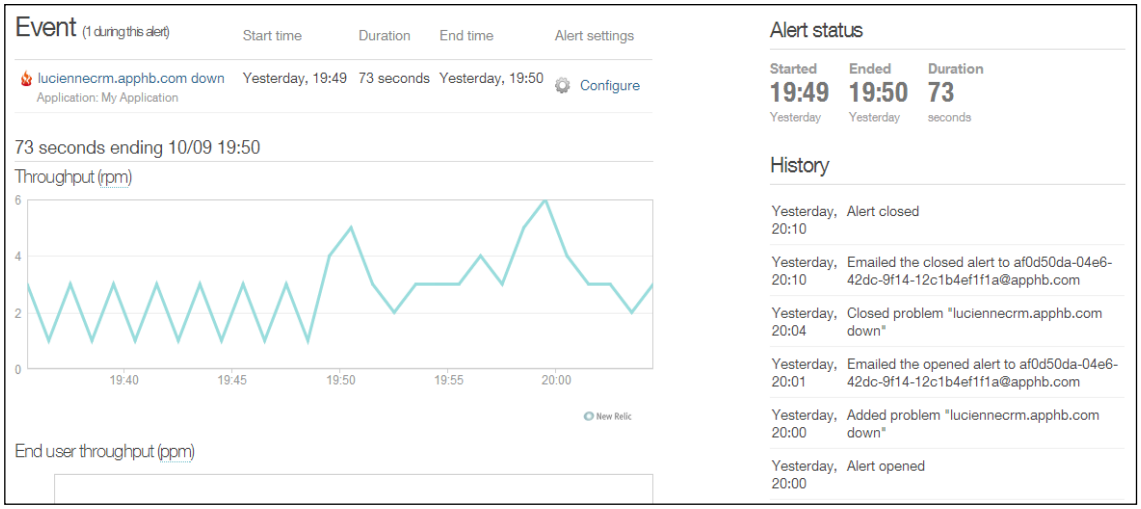


FIGURE 13 - THIS IS A SAMPLE REPORT ISSUED BY NEW RELIC. THIS GRAPH SHOWS A CONSTANT THROUGHPUT TIME UNTIL 19.40. AT 19.40 THE SYSTEM WAS DOWN FOR 73 SECONDS AS DISPLAYED IN THE GRAPH. THE LIST ON THE RIGHT SHOWS THE HISTORY DURING THE 3-MINUTE DOWNTIME.

4.4.2 GOOGLE ANALYTICS

The aim of Google Analytics in this report was to gather analytics related to the website and performance. Although New Relic was tracking similar data, the idea was to have another tool to double-check statistics of the same application.

The process to set up the application on Google Analytics consisted of a registration at the Analytics account through http://www.google.com/analytics. The tracking process started when the URL was added to the Analytics page and the following code was added to <head> section of _Layout.cshtml in the Shared View Folder of the MVC application.

```
<script type="text/javascript">
        (function (i, s, o, g, r, a, m) {
            i['GoogleAnalyticsObject'] = r; i[r] = i[r] || function () {
                (i[r].q = i[r].q || []).push(arguments)
            }, i[r].l = 1 * new Date(); a = s.createElement(o),
            m = s.getElementsByTagName(o)[0]; a.async = 1; a.src = g;
m.parentNode.insertBefore(a, m)
         })(window, document, 'script', '//www.google-
analytics.com/analytics.js', 'ga');

        ga('create', 'UA-42941775-1', 'apphb.com');
        ga('send', 'pageview');
    </script>
```

Figure 14 is snapshot of the data gathered and presented by Google Analytics:

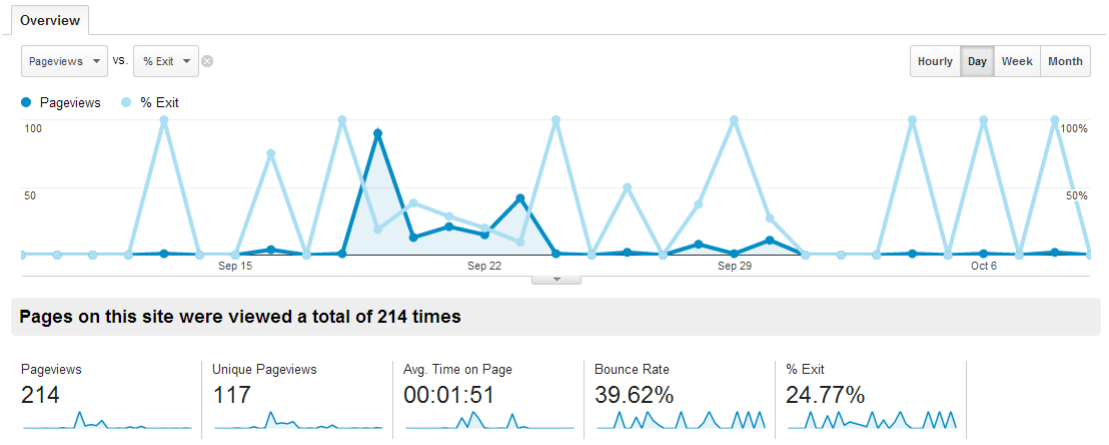


FIGURE 14 – A SAMPLE OF DATA PROVIDED BY GOOGLE ANALYTICS. THE GRAPH ABOVE DISPLAYS THE NUMBER OF PAGE VIEWS AND ESTIMATED TIME TO LEAVE THE PAGE. THE BOUNCE RATE IS ESTIMATED TIME VIEWERS STAYED ON THE PAGE.

4.4.3 Code Viewer Metrics

The Code Viewer Metrics tool was integrated into Microsoft Visual Studio to calculate the following metrics: Maintainability Index, Class Coupling, Cyclomatic Complexity, Depth of Inheritance and Lines of Code. The Code Metrics Viewer required an installation through this link: http://bit.ly/iipJuw and when complete, the installation path of the Code Metrics Power Tool was set up as shown in *Figure 15*:

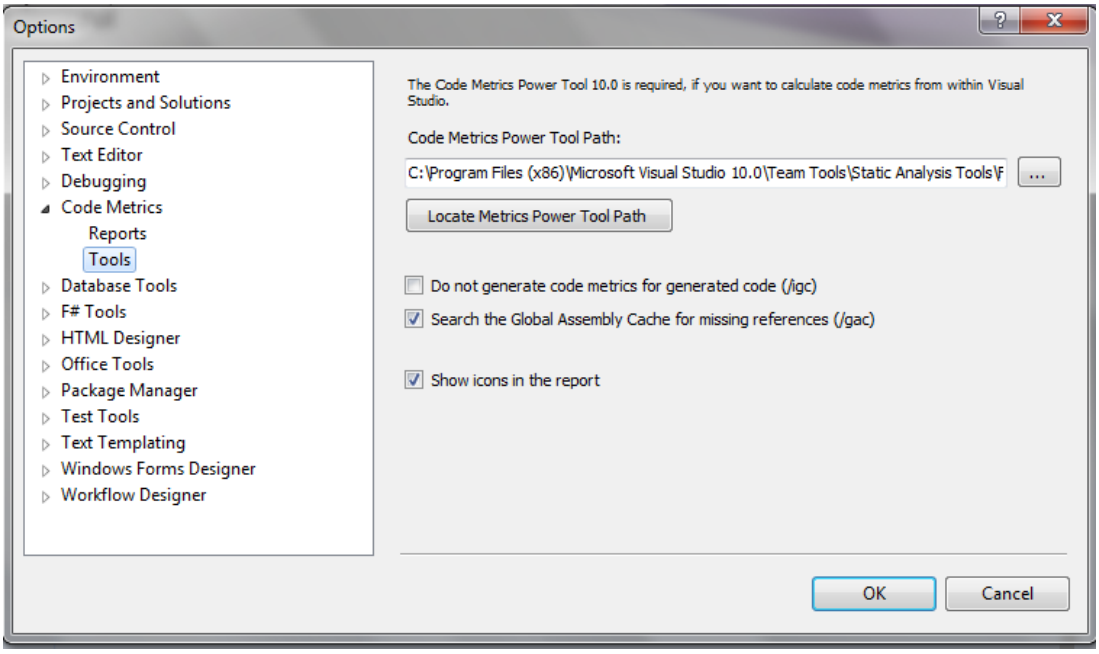
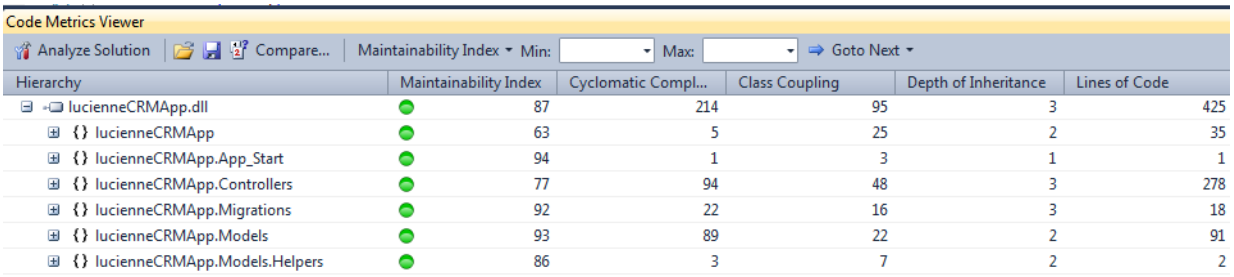


FIGURE 15 - CONFIGURATION SETTINGS OF CODE METRICS VIEWER. THE PURPOSE OF THE CODE METRICS TOOL PATH IS TO BE ABLE TO RUN HE METRICS EXECUTABLE FILE WHICH IS FOUND IN THIS PATH

Metrics were generated by clicking the 'Analyse Solution' button. *Figure 16* shows the metrics of the CRM application. Low values for cyclomatic complexity, class coupling, depth of inheritance and lines of code are indicative of better quality software. On the other hand, high values of maintainability index indicate that software is highly maintainable.



Hierarchy	Maintainability Index	Cyclomatic Compl...	Class Coupling	Depth of Inheritance	Lines of Code
lucienneCRMApp.dll	87	214	95	3	425
{} lucienneCRMApp	63	5	25	2	35
{} lucienneCRMApp.App_Start	94	1	3	1	1
{} lucienneCRMApp.Controllers	77	94	48	3	278
{} lucienneCRMApp.Migrations	92	22	16	3	18
{} lucienneCRMApp.Models	93	89	22	2	91
{} lucienneCRMApp.Models.Helpers	86	3	7	2	2

FIGURE 16. THE METRICS OF CODE METRICS VIEWER CALCULATES THE MAINTAINABILITY INDEX (MI), CYCLOMATIC COMPLEXITY(CC), DEPTH OF INHERITANCE (DIH), CLASS COUPLING(CLC) AND LINES OF CODE (LOC). MI IS AN OVERALL-QUALITY INDICATOR AND IS CALCULATED USING HALSTEAD VOLUME. CC CALCULATES THE CONTROL-FLOW GRAPH BY DETERMINING HE NUMBER OF BRANCHES AND THEIR DEPENDENCIES. DIH INDICATES THE NUMBER OF BASE CLASSES. LOC METRIC DEPENDS ON THE INTERMEDIATE LANGUAGE GENERATED BY THE COMPILER.

4.4.5 NuGet Packages used

NuGet consists of several reusable packages that may be installed on a Microsoft Framework. Table 5 lists the packages installed and used for this application.

NuGet Package	Purpose
EntityFramework	This package enabled easier access to manage relational data in a database.
Google.Apis	This was used as a runtime client when using Google Services, such as downloading photos from Google Drive in the download section, integrating a Google Map in the contact us section and a form to send email to Gmail.
JQuery	This package was used to handle simple Jquery photo animation on the homepage
Metrics	This package was used to capture CLR and application-level metrics.
Microsoft.AspNet.SignalR	A package was used to create real-time chat room.
Microsoft.Net.Http	This package was used to process HTTP requests and responses.
MvcScaffolding	This was useful to add views, controllers and models to the MVC application

TABLE 5 - NUGET PACKAGES. THIS TABLE LISTS ALL THE PACKAGES USED IN THIS APPLICAITON AND THEIR PURPOSE

4.5 Software Process Model for SaaS

The software process model used to design and implement the SaaS was the Agile Methodology. The Agile Methodology was used because as stated in the Agile manifesto it facilitates collaboration between the software developer and the customer and embraces change. I believe that these are the main factors that contribute to a successful project. During the development phase I uploaded the source code on subversion Codeplex and released it on the public cloud Appharbor. This process adopted the SCRUM framework. Every sprint consisted of a release of two or three objectives and the average timeframe was one week.

A breakdown of the agile software process used is illustrated in *Figure 17:*

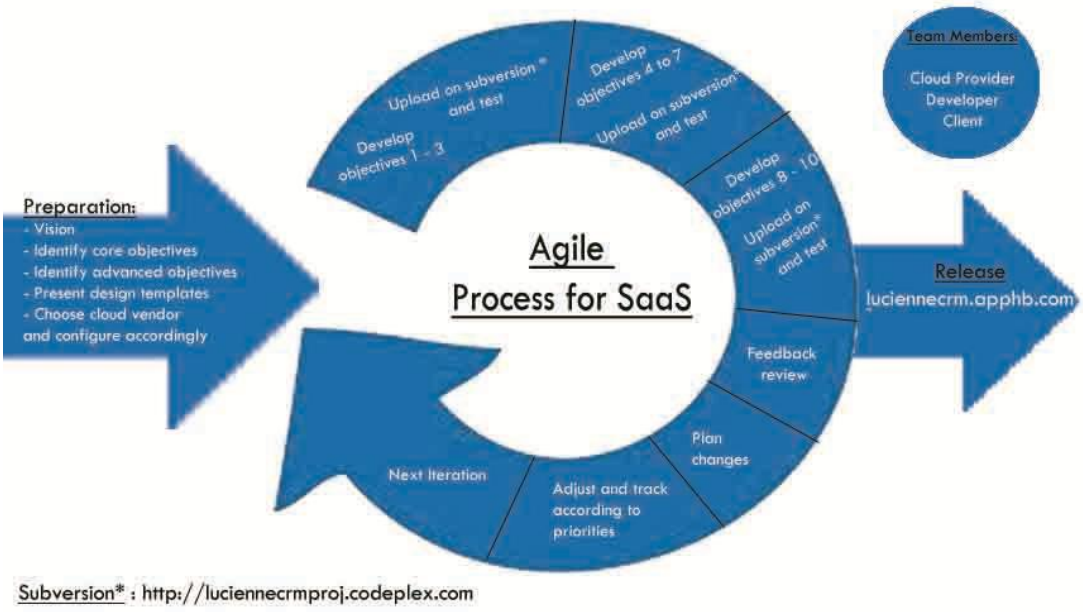


FIGURE 17 - THE AGILE PROCESS MODEL FOR SAAS. THIS AGILE PROCESS WAS USED WHEN DEVELOPING THIS PROJECT. THE TEN OBJECTIVES WERE IMPLEMENTED IN SETS OF THREE, REVIEWED, UPDATED, TESTED, UPLOADED ON SUBVERSION AND DEPLOYED TO APPHARBOR

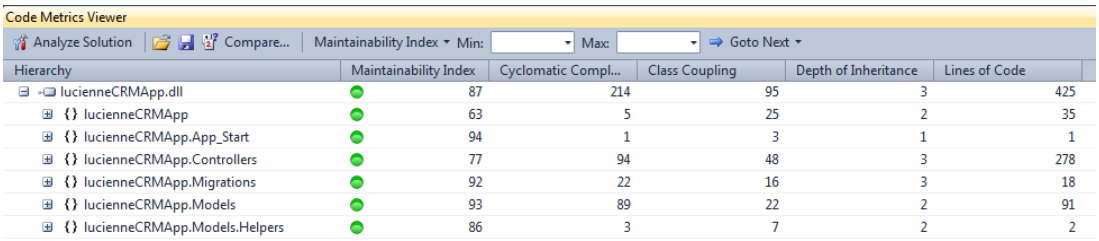
This project commenced with a list of ten objectives which were implemented in sets of three, uploaded to subversion and tested. The purpose of uploading to subversion was for the client to be able to download and access the latest version of the project as well as to ease client-developer communication. On completion of the ten objectives, the project was released to Appharbor where the user could test the application and in the process metrics were being gathered. The project was updated and modified until the clients' requirements were met.

CHAPTER 5 - EVALUATION OF RESULTS:

This chapter presents an evaluation of results generated from third-party software; New Relic, Google Analytics and Code Viewer Metrics (Visual Studio). Results were collected between 13th August and 30th November 2013.

5.1 REUSABILITY

The purpose of developing a model-view-controller (MVC) architecture pattern, for this project, was to ensure reusability of code. The *Figure 18* shows that this project consisted of 425 lines of code.


Code Metrics Viewer

Analyze Solution | Compare... | Maintainability Index Min: Max: Goto Next

Hierarchy	Maintainability Index	Cyclomatic Compl...	Class Coupling	Depth of Inheritance	Lines of Code
lucienneCRMApp.dll	87	214	95	3	425
{} lucienneCRMApp	63	5	25	2	35
{} lucienneCRMApp.App_Start	94	1	3	1	1
{} lucienneCRMApp.Controllers	77	94	48	3	278
{} lucienneCRMApp.Migrations	92	22	16	3	18
{} lucienneCRMApp.Models	93	89	22	2	91
{} lucienneCRMApp.Models.Helpers	86	3	7	2	2

FIGURE 18. METRICS OF CODE METRICS VIEWER. MI IS AN OVERALL-QUALITY INDICATOR AND IS CALCULATED USING HALSTEAD VOLUME. CC CALCULATES THE CONTROL-FLOW GRAPH BY DETERMINING HE NUMBER OF BRANCHES AND THEIR DEPENDENCIES. DIH INDICATES THE NUMBER OF BASE CLASSES. LOC METRIC DEPENDS ON THE INTERMEDIATE LANGUAGE GENERATED BY THE COMPILER.

The Maintainability Index is considered to be an overall-quality indicator and was calculated as follows:

$$\textit{Maintainability Index} = 171 - 5.2 * \ln(\textit{Halstead Volume}) - 0.23 * (\textit{Cyclomatic Complexity}) - 16.2 * \ln(\textit{Lines of Code})$$

EQUATION 1 - MAINAINABILITY INDEX USED BY CODE METRICS VIEWER

The maintainability Index of this project is between 63 and 92. These figures fall within the acceptable range, according to MSDN.

5.2 RELIABILITY

Figure 19 below illustrates the error rate in percentages during the period from 13th August to 30th November.

The highest error rate recorded was on 19th September (0.0014%). During that week the response time was between 16.4ms and 1780ms. In fact, error reports showed that that on 20th September the system was down twice, for 7minutes and 8 minutes respectively; this was the highest downtime period recorded.

On 28th August, the error rate was 0.0013% and the response time was double the average at 43.8ms. On 23rd November, the error rate was higher than average at 0.0011% and the response time was 72.2ms.

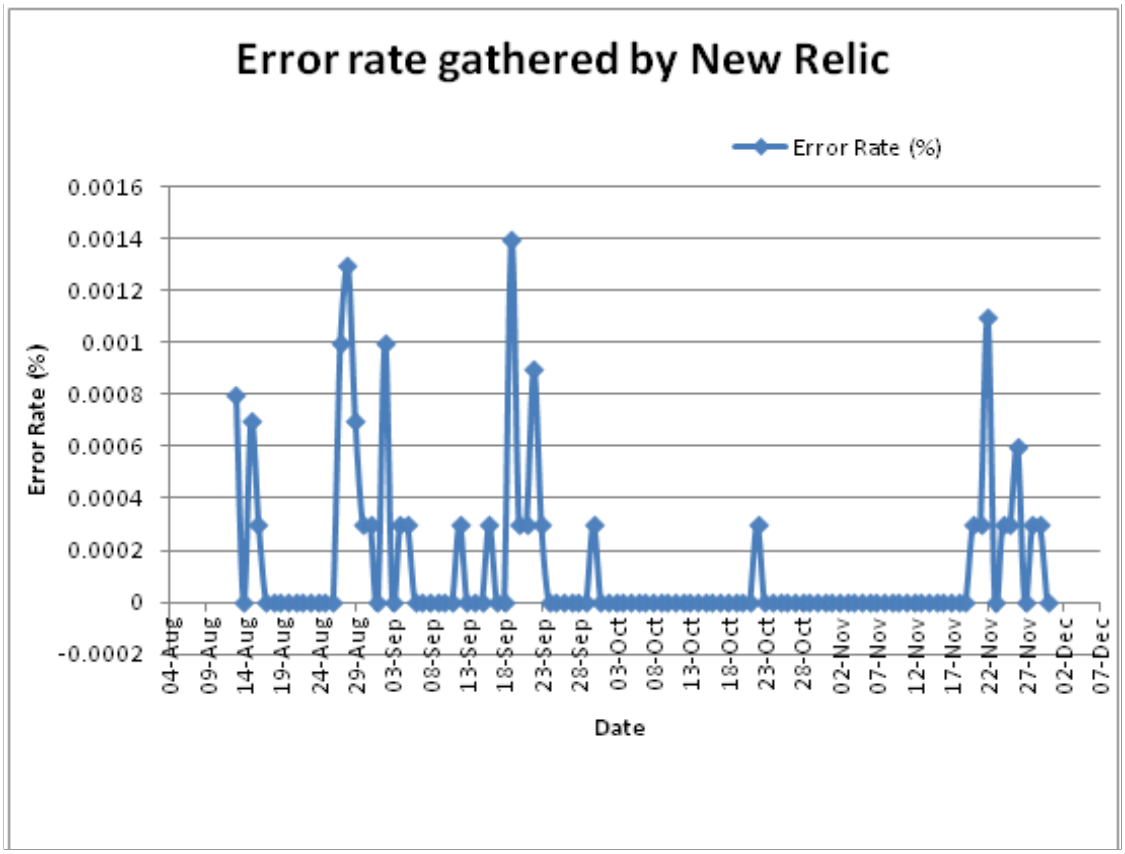


FIGURE 19 - ERROR RATE RECORDED BY NEW RELIC. THE GRAPH ABOVE SHOWS THE ERROR RATE IN PERCENTAGES DURING THE TESTING PHASE OF THIS PROJECT. RESULTS SHOW THAT ON 28TH AUGUST, 19TH SEPTEMBER AND 23RD NOV THE SYSTEM RECORDED THE HIGHEST ERRORS.

5.3 SCALABILITY

To be able to measure the scalability, the throughput (rate per minute) was correlated with the time to load in seconds. The highest response time was recorded between 21st September and 23rd September, marking an increase of 72% of the average response time gathered during this testing phase. The throughput rate was also higher with an average increase of 1 minute.

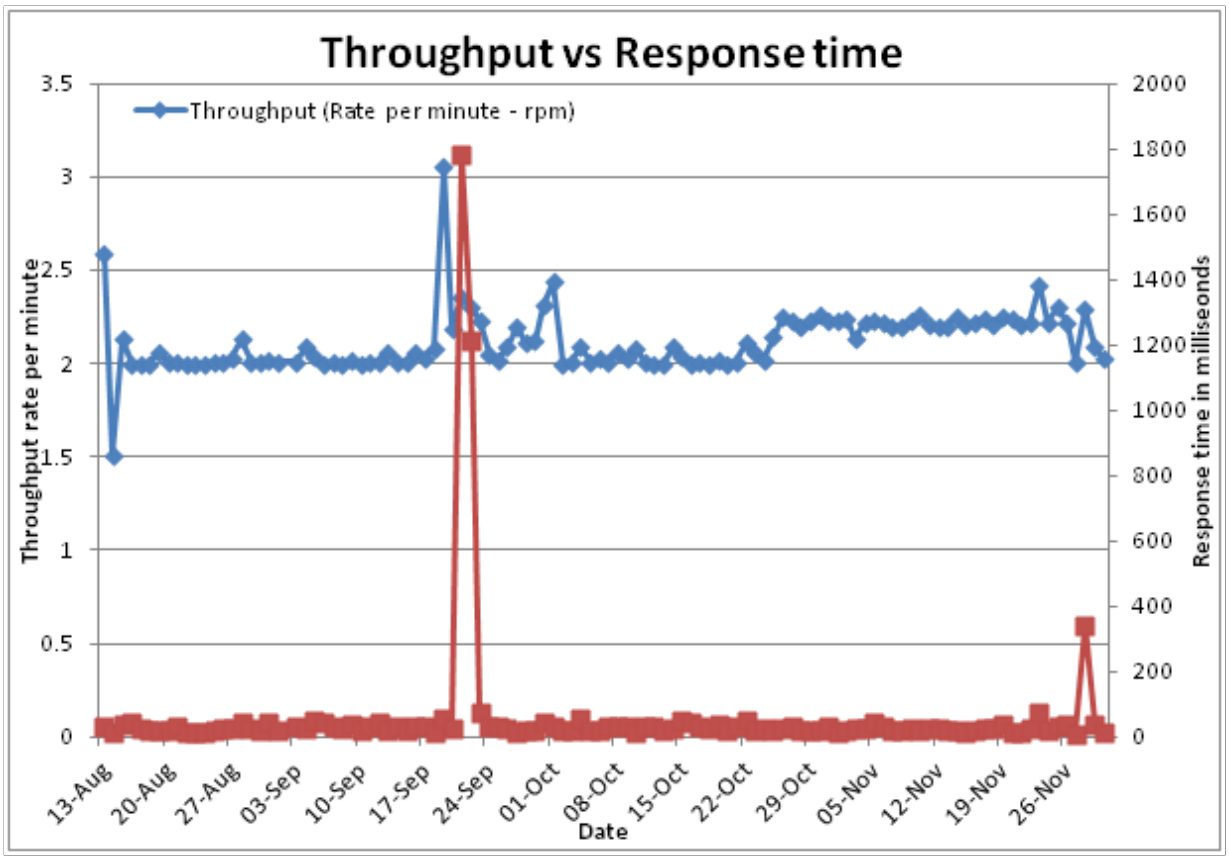


FIGURE 20 - THROUGHPUT RATE VS RESPONSE TIME RECORDED BY NEW RELIC. THIS CORRELATION SHOWS THAT WHEN THE RESPONSE TIME WAS HIGH BETWEEN 21ST AND 23RD SEPTEMBER, THE THROUGHPUT RATE INCREASED, THUS SCALABILITY WAS LOW.

As a result, the system was unstable for 3 days. During the remaining testing phase, results show that requests have been shared across servers without affecting the application.

5.4 AVAILABILITY

The availability metric was calculated using this formula provided by Oracle (2012).

Availability = Average Time to Failure (ATTF) / (average time to failure (ATTF) + average time to recover (ATTR))

EQUATION 2 - AVAILABILITY RATE PROVIDED BY ORACLE (2012)

Table 6 shows the days when the system failed and the average time it took to recover. On 19th September and 20th September the system recorded the lowest availability day as the system failed more than once during those days. On the other days no downtime was recorded.

Date	Average Time to Failure (in minutes)	Average time to recover (in minutes)	Availability (%)
19-Sep	32	18	64%
	150	5	97%
	182	3	98%
	7	3	70%
	22	2	92%
	82	12	87%
	105	7	94%
20-Sept	17	4	81%
	22	8	73%
04-Oct	56	25	69%

TABLE 6 - SYSTEM DOWNTIME DATE RECORDED BY NEW RELIC. THIS TABLE SHOWS THAT ON 19TH SEPTEMBER THE SYSTEM WAS DOWN FOR 7 CONSECUTIVE TIMES WITH AN AVERAGE AVAILABILITY RATE OF 86%. ON 20TH SEPTEMBER THE SYSTEM WAS DOWN TWICE AN ON 4TH OCTOBER THE SYSTEM WAS DOWN FOR 25 MINUTES, RESULTING IN AN AVAILABILITY RATE OF 69%

Although the system was unstable for 3 days, results generated by New Relic, as can be seen in *Figure 21*, show that the application provided an overall of 99.988% availability.

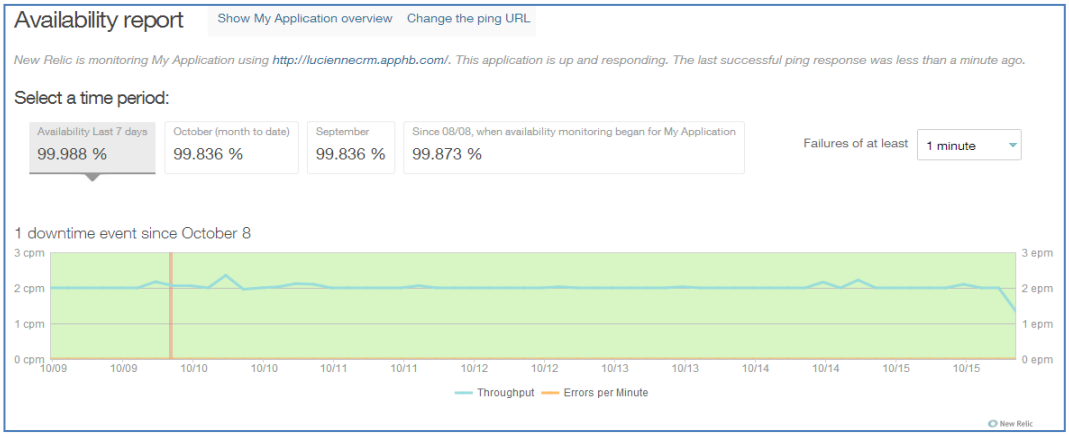


FIGURE 21 - AVAILABILITY REPORT PROVIDED BY NEW RELIC. ALTHOUGH THE SYSTEM WAS UNSTABLE FOR 3 DAYS DURING THE TESING PHASE, THIS GRAPH SHOWS THAT THE AVAILABILITY RATE WAS OF 99.988%.

5.5 PERFORMANCE (EFFICIENCY)

To be able to analyse whether hosting on a public cloud would affect the performance, the server request time and response time were monitored across four European cities: London, Rotterdam, Stockholm and Strasbourg.

Data was generated through a free online application hosted by Alertra.com. The common variables across all networks included Content Length (equal to 6.46Kb) and Data Size (equal to 7.35Kb).

Table 7 presents the Transfer time and Response time from various satellites:

Location	Request time	Transfer time
London	0.060458 Sec	0.002352 Sec
Rotterdam	0.084044 Sec	0.003581 Sec
Stockholm	0.185254 Sec	0.003308 Sec
Strasbourg	0.559113 Sec	0.007634 Sec

TABLE 7 : RESULTS GENERATED BY ALERTRA. THIS TABLE SHOWS THE AVERAGE REQUEST TIME AND TRANSFER TIME, FROM VARIOUS LOCATIONS, DURING THE TESTING PHASE OF THIS PROJECT.

As illustrated in *Figure 22*, the time to transfer was similar in all locations but the request time varied in the case of one location. The difference was negligible at 0.4 seconds and the system was not affected.

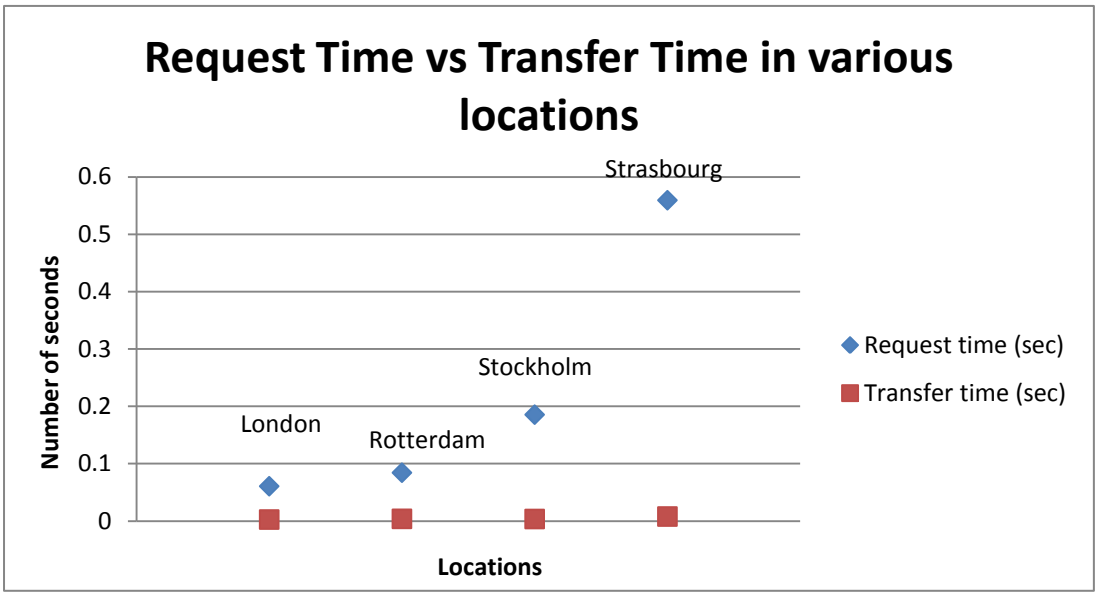


FIGURE 22 – REQUEST TIME VS TRANSFER TIME RECORDED BY NEW RELIC. REQUEST TIME AND TRANSFER TIME IN SECONDS ARE RECORDED ON THIS GRAPH. STRASBOURG SHOWS A 0.4 SECOND DIFFERENCE FROM STOCKHOLM. WHEREAS THE REQUEST TIME AND TRANSFER RATE IN LONDON AND ROTTERDAM WHERE SIMILAR.

CHAPTER 6 - DISCUSSION:

In this chapter, I shall be discussing the three main factors of a public cloud: metrics, cost and security in relation to the results as detailed in the previous chapter.

6.1 METRICS

Results show that the system was available for 99.988% during the testing phase. The system encountered downtime on three consecutive days between 19th September and 21st September. Qian H. et. Al (2011) and Avizienis A. et Al (2004) stated that the main factors that contribute to failure are: software failures, network failures, power failures and server failures. Cloud Alliance Security in their report suggested that cloud-related malware may also contribute to failure. In view of the fact of having an application deployed on a public cloud and used for research purposes only, the downtime period did not have a huge impact. When analysing the downtime report provided by New Relic to identify the source of the downtime, I noted very limited detail in the report as shown in *Figure 23*.

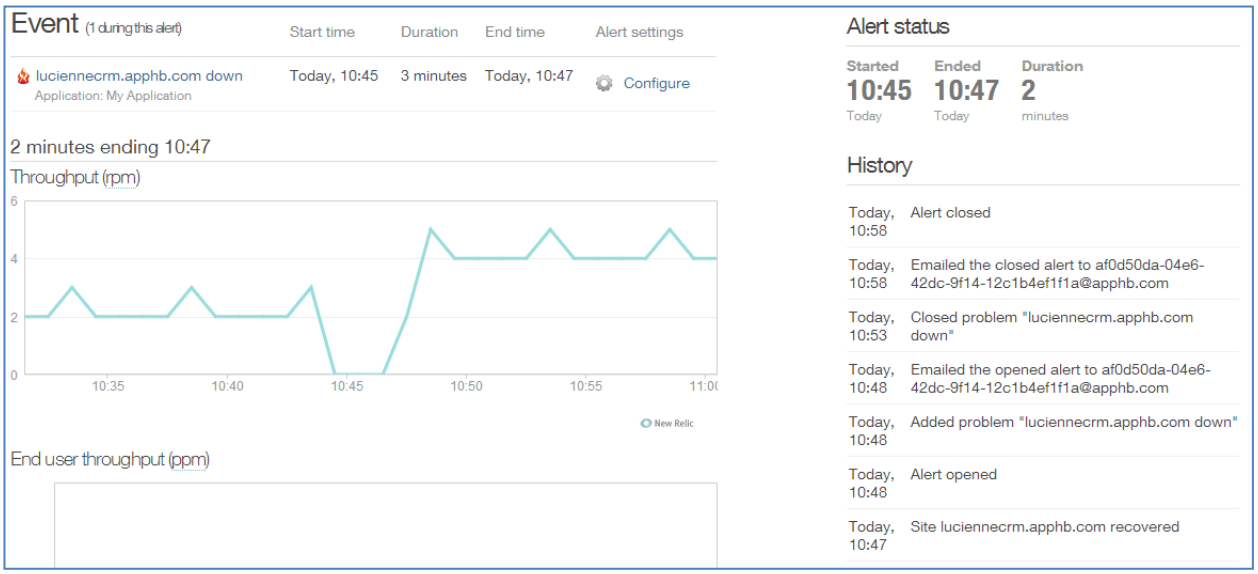


FIGURE 23. THIS IS A REPORT ISSUED BY NEW RELIC SHOWING THE THROUGHPUT TIME. AT 10.45 THE SYSTEM WAS DOWN FOR 3 MINUTES AS DISPLAYED IN THE GRAPH. THE LIST ON THE RIGHT SHOWS THE HISTORY DURING THE 3-MINUTE DOWNTIME.

Unfortunately, it was impossible to detect whether the cause of the downtime was a result of a software, network, power or server failure. In my opinion, providing detailed error logs will help the developer to minimize errors.

Quoting Miller (1968) and Card et al. (1991), Nielsen J. (1993) mentions three main response time thresholds regarding efficiency, namely, 0.1s, 1s or 10s. When the response time is within 0.1s, users view the application as reacting instantly. The limit users are able to wait for an application to respond is between 1s and 5s. When the response time is longer than 10s, users become unfocused and usually perform other tasks while waiting. When comparing these thresholds to the results obtained in this report, the response time varied slightly in different locations. Results show that the widest discrepancy was of 0.4s, confirming a highly positive outcome.

6.2 COST

For this report I availed myself of the free scheme provided by AppHarbor. This project, therefore, was very cost effective. Although I did not pay for the service of the cloud provider, the customer service was very efficient and all my queries were answered in detail within 24 hours. Since AppHarbor also includes a list of plugins for its customers, I was able to benefit from the following plugins for free: New Relic, SQL Server and AirBrake.

6.3 SECURITY

Although cloud computing has been around for a long time, a grey area still exists concerning the security of software in the cloud. A report submitted by *Cloud Security Alliance* states that Google reported the highest number of attacks, followed by Amazon and Microsoft. To be able to investigate the security and privacy policy of AppHarbor, a thorough analysis of the Privacy Policy, Terms of Service and DMCA (Digital Millennium Copyright act) were carried out.

In the Privacy Policy section, AppHarbor guarantees that in case of third party trespassing security measures and managing to access private communications, the company will "post a reasonable prominent notice" to the clients' websites. In my opinion, breaching of data should be given more importance and clients should be notified immediately via private messages.

AppHarbor states that this company adopts the US privacy policy law since it is hosted in the US. This implies that, although I am accessing my application from a European Union country, I am not entitled to benefit from the Privacy Policy of the EU.

As mentioned in Section 3.6 Challenges, the EU privacy law states that customers have the right to complain if their personal data was misused whereas the US privacy law (1974) states that a person has the right to view personal data, ask for amendments and be informed of any disclosures. The US privacy law does not specifically state that in case of breach the customer has the right to complain to the relevant authorities. In my opinion, the privacy law in the US should be updated to take into account the latest technological infrastructures and bring it closer to the EU privacy law.

AppHarbor states that the company has "implemented reasonable measures" but it does not guarantee that these measures eliminate security risks completely and it does not imply that they are adopting unique security measures. In its Terms of Services, it is specifically stated that the clients are ultimately responsible for the security of their passwords and personal account. Incidentally, during the writing of this report, as a client of AppHarbor, I received an email informing me about a security breach relating to MongoHQ which is one of the database clients. This incident is annexed as Appendix E, *Security Breach Notification by AppHarbor*. It is certainly important that AppHarbor informs all its clients, including those who are not using MonogHQ, when breaches of this kind occur. This information should be delivered instantly. In my case, the notification was sent on 2nd November when the breach occurred between 27th October and 29th October.

Oracle (2012) suggests that when choosing a cloud vendor, one should ensure that the cloud vendor meets the general and industry-specific security and compliance standards such as those established by the Payment Card Industry (PCI) Security Standards Council or the National Institute of Standards and Technology (NIST). It should be noted that AppHarbor does not make any reference to these standards.

Summing up, one can say that this web application was hosted for free and although there were three days when the system was unstable, work on this project could still be carried out. However, it should be noted that three days of instability would have a major negative impact on systems used by commercial, educational or healthcare institutions. As has been pointed out already, the security standards and privacy policies provided by AppHarbor do not guarantee maximum security measures.

CHAPTER 7 - CONCLUDING REMARKS

This chapter consists of a recapitulation of this study, evaluation of the whole project, limitations and identification of areas where further work is needed.

7.1 - RECAPITULATION

The objectives set for this research project have been achieved. In fact, an application deployed on a public cloud has been implemented, runtime quality metrics; reusability, reliability, scalability, availability and efficiency have been evaluated, costs have been assessed and security attributes have been analysed.

The present study shows that for a personal and a small-size application it is cost effective to deploy on a public cloud such as Appharbor. As explained in Chapter 3.6, it is cheaper to deploy on a public cloud than on a private server. Besides, it is also possible to deploy for free on Appharbor, as it provides for one free worker. The analysis of the quality metrics shows that the system has been reliable and highly available, while the resources have been managed efficiently. As a result, the integration of Web 2.0 features such as instant messaging, Google drive, Google maps and Facebook did not have an impact on the system.

With reference to security, this study confirms what has emerged from the literature review, namely, that cloud providers are still struggling to provide 100% guarantee, even though they generally claim they have invested in security infrastructures to prevent intruders. In fact, when analysing the security infrastructure of this application deployed on AppHarbor, the application was not attacked, during the testing phase but a notification came regarding a security breach to MongoHQ, which is one of AppHarbor's client. The detailed notification is annexed as *Appendix E – Security Breach Notification by AppHarbor*. The report noted that the breach was related to direct database access which involved access to MongoHQ accounts database and retrieval of personal information. This confirms the findings published by Khalil, Issa M. et. Al (2013) that there are forty-six issues related to security that have not been addressed as yet. Security in cloud computing is clearly a concern across all types of clouds: public, private and hybrid. In my opinion, this is the reason why some companies are still hesitant about storing all their data and managing it on the cloud.

It is likely that more companies would migrate to the cloud, if cloud providers try to implement security measures in a meticulous way. A possible solution may be that

companies ensure that their data is encrypted and that they choose a reliable cloud provider that guarantees security and a reliable SLA. Companies can also opt for a hybrid system whereby owners would choose which applications would be managed by the company's servers and which would be managed by the cloud.

As illustrated in *Figure 4*, the role of the cloud providers is essential in the software engineering process when developing SaaS. The involvement of other parties in a team may contribute to three constraints: duration of the project, cost and communication. The third constraint is self-explanatory, as the more members involved in the team the more difficult it is to communicate and collaborate especially if members are located in different time zones. The project might be slightly longer to be in a position to include the selection process of the cloud provider and the technical configurations to deploy on the cloud. As a result, this may contribute to additional costs that were not included in the traditional process.

In conclusion, one may say that the storage models and features of the cloud providers researched in this report may differ but the pricing models and infrastructure are very similar. In my opinion, the competition between cloud providers is increasing and focusing on the run-time quality attributes and cost-efficient pricing plans rather than improving SLAs and security measures. Hence, when choosing a cloud provider, companies should take into consideration not only performance and pricing plans but also the quality of service and security measures.

7.2 - PROJECT EVALUATION

Through this project I acquired several soft skills. I learned how to organise ideas, plan and manage the technical application and compile the final report. The literature review helped me to identify the relevant sources and analyse them in a critical way. The project plan, written for the Extensive Project Proposal (EPP), served as a useful guide not only to keep to the stipulated time frames but also and, in particular, to develop a consistent line of thinking.

This project proved useful in so far as technical skills are concerned. At the beginning, I decided to build the application using C# because of familiarly with the programming language. This choice gave me more time to explore and make use of extensive libraries that I had never used before. Although we were free to choose the

programming language of our choice, I regret that I was unable to use PHP. This language is being used widely by developers nowadays and it would have been provided me with a good opportunity to gain more knowledge. Unfortunately, time did not permit but ultimately the programming language used was not of any particular relevance to the present project and findings.

The EPP was due a month after this module commenced. Although I had started my research well ahead of the date of submission, I had not considered carefully enough the implications of one of the advanced objectives. This objective was supposed *to ensure energy efficiency and sustainability for software on the cloud*. After receiving the feedback sent by my second tutor, I realized that this objective includes extensive research and may be a project in itself. For this reason, I decided not to include this objective in the present report but I made reference to it in section 7.4.1 Energy efficiency and sustainability for software in the cloud

Another matter to which I had not given sufficient thought in the EPP concerned the collection of metrics. In the EPP I presented the formulas as proposed by several authors. However, following research and experiments I found that third party plugins such as: New Relic, Alertra and Google Analytics could be usefully used in my project to gather real-time data. In fact, the integration of third party plugins ensured more accurate results than working out statistical formulas.

In general, I am satisfied with the final outcome, as I managed to finish the report in time and succeeded in answering the research questions and in achieving all the objectives. This project has opened for me new opportunities for exploring an area of research in cloud computing that is very important both from a technical and an ethical perspective and is yet not sufficiently researched.

7.3 - LIMITATIONS

The main constraints in this project were cost, time and number of participants. Due to cost limitations this research could not be extended to different cloud vendors and show by a comparison between various cloud providers whether quality improves when paying for a service. Time constraints did not allow the present researcher to study other programming languages such as Ruby or PHP. If a longer time was available, further data could be collected and used to bring out a wider picture of

the matter under study. The last limitation of this project was the number of participants. The scalability results emerging from this study are reliable but they could be enhanced by the participation of more participants.

7.4 - FUTURE WORK

In the course of this report, the following two areas related to cloud computing were identified as requiring further investigation.

7.4.1 ENERGY EFFICIENCY AND SUSTAINABILITY FOR SOFTWARE IN THE CLOUD

Koomey J. (2010) states that the electricity consumption by data centres in 2010 is about 1.3% of all electrical energy worldwide and 2% of the electricity consumed in the US. Priya, B. et. Al (2013) describe how cloud computing provides a solution as part of the Green IT initiative to promote environmental responsibility. Priya, B. et. Al (2013) and Avizienis A. (2004) explain that in cloud computing, server virtualisation reduces the total physical server footprint, since the server load can be migrated to available servers, sharing excess storage among a large number of virtual machines. The authors mention how cloud-based infrastructure relies on automation software, thus allowing developers to utilise high ratios of cloud resources, and in the process maximise energy and resource efficiency. Pay-per-use and Self-service methods also contribute to an energy efficient cloud.

The urgency of using energy efficiently in today's digital world demands that programmers should develop environmentally friendly software.

7.4.2 APPLYING MORE SECURITY STANDARDS AND MEASURES FOR SOFTWARE IN THE CLOUD

As has been pointed out in this report, cloud vendors are seeking to optimize the software, upgrade the service and lower the cost rather than improve security. For example, Appharbor has more than three plugins available to monitor the performance of the application and another three plugins to monitor errors. The majority of the research journals analysed in this report mentioned that security in the cloud is a challenge still to be addressed. As a matter of fact, Sabahi F. (2011) states

that companies prefer to store less personal information on the cloud and store sensitive information on the local servers.

A Service Level Agreement (SLA), guaranteeing maximum security measures, would certainly provide a really good quality service, as this is a matter of concern for clients when choosing a reliable cloud provider.

REFERENCES:

Alford T. & Morton G. (2010). '*The Economics of Cloud Computing' [*pdf] Available at http://www.boozallen.com/media/file/Economics-of-Cloud-Computing.pdf [Accessed 10 August 2013]

Alhamad M., Dillon T., Wu C., Chang E. (2010). '*Response Time for Cloud Computing Providers.*' WAS2010, France, (8-10 November 2010).

Appharbor, N/A. '*Privacy Policy*' [Online] Available at: https://appharbor.com/page/privacypolicy [Accessed 29th October 2013]

Avizienis A., Laprie J.C. , Randell B., Landwehr C. (2004). '*Basic Concepts and Taxonomy of Dependable and Secure Computing*', IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, pp. 11–33, Jan. 2004.

Baklizi, M.; Alghyaline, S (2011). '*Evaluation of E-Learning websites in Jordan universities based on ISO/IEC 9126 standard*', Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, On page(s): 71 – 73

Behl, A. (2011). '*Emerging security challenges in cloud computing: An insight to cloud security challenges and their mitigation*,' Information and Communication Technologies (WICT), 2011 World Congress on , vol., no., pp.217,222, 11-14 Dec. 2011

Boehm, B.W. (1976), '*Software Engineering*', Computers, IEEE Transactions, vol.C-25, no.12, pp.1226,1241, Dec. 1976

Brandon J. (2013). '*Cloud providers need to work on SLA's and reliability, says NATS CIO | Business Cloud News. 2013. Cloud providers need to work on SLA's and reliability, says NATS CIO | Business Cloud News.*' [ONLINE] Available at: http://www.businesscloudnews.com/2013/07/04/cloud-providers-need-to-work-on-slas-and-reliability-says-nats-cio/ . [Accessed 15 September 2013].

Card, S. K., Robertson, G. G., and Mackinlay, J. D. (1991). '*The information visualizer: An information workspace*'. Proc. ACM CHI'91 Conf. (New Orleans, LA, 28 April-2 May), 181-188.

Chidamber, S.R.; Kemerer, C.F. (1994). '*A metrics suite for object oriented design*', Software Engineering, IEEE Transactions on Software Engineering , vol.20, no.6, pp.476,493, Jun 1994

Chieu, T.C.; Mohindra, A.; Karve, A.A. (2011), '*Scalability and Performance of Web Applications in a Compute Cloud*', e-Business Engineering (ICEBE), 2011 IEEE 8th International Conference on , vol., no., pp.317,323, 19-21 Oct. 2011

Chiew, T. K., (2009). '*Web page performance analysis*'. PHD. Glasgow: University of Glasgow.

Cloud Security Alliance, (2013). '*Cloud Computing Vulnerability Incidents: A Statistical Overview.*'[pdf] Available at:http://www.cert.uy/wps/wcm/connect/975494804fdf89eaabbdab1805790cc9/Cloud_Computing_Vulnerability_Incidents.pdf?MOD=AJPERES [Accessed 10 August 2013]

CloudTweaks, (2013). '*Effects Of Hardware Problems In Cloud Based Systems*' [Online] Available at: http://www.cloudtweaks.com/2013/07/effects-of-hardware-problems-in-cloud-based-systems/ [Accessed 22 09 2013].

Consortium for IT Software Quality, (2013). '*CISQ Standards*' [Online]. Available at: http://it-cisq.org/standards-page/ [Accessed 17th August 2013]

Cunsolo, V.D.; Distefano, S.; Puliafito, A.; Scarpa, M. (2010). '*Applying Software Engineering Principles for Designing Cloud@Home, Cluster, Cloud and Grid Computing (CCGrid)', 2010 10th IEEE/ACM International Conference* , vol., no., pp.618,624, 17-20 May 2010

Curry, R.; Kiddle, C.; Markatchev, N.; Simmonds, R.; Tingxi Tan; Arlitt, M.; Walker, B. (2008), '*Facebook Meets the Virtualized Enterprise*, Enterprise Distributed Object Computing Conference, 2008. EDOC '08. 12th International IEEE , vol., no., pp.286,292, 15-19 Sept. 2008

Da Silva E. & Lucredio D, (2012), '*Software engineering for the cloud: a research roadmap*', 2012 Brazilian Symposium on Software Engineering.

Dai, Y. S., Xie, M., and Poh, K. L. (2006). '*Reliability of grid service systems.*' Computers and Industrial Engineering, Vol. 50, pp. 130-147

Dai, Y. S., Yang, B., Dongarra, J., & Zhang, G. (2009). '*Cloud service reliability: Modeling and analysis*'. In The 15th IEEE Pacific Rim International Symposium on Dependable Computing.

Department of Justice United States, (2012). '*OVERVIEW OF THE PRIVACY ACT OF 1974*'.[pdf] Available at: http://www.justice.gov/opcl/1974privacyact-2012.pdf [Accessed on 29th October 2013]

Draheim, D., (2012). '*CASE 2.0: on key success factors for cloud-aided software engineering*'. New York, ACM.

Erl, T., Mahmood, Z. & Puttini, R., (2013). '*Cloud Computing: Concepts, Technology and Architecture.*' 1st ed. United States: Prentice Hall.

Erni, K.; Lewerentz, C., (1996). '*Applying design-metrics to object-oriented frameworks,*' Software Metrics Symposium, 1996., Proceedings of the 3rd International, vol., no., pp.64,74, 25-26 Mar 1996

Esparza-Peidro J. & Munoz-Escoi F. D. (2011). '*Towards the next generation of model driven cloud platforms*'. Preceedings of the 1st international conference on cloud computing and services science, Closer 2011, pp. 494-500

European Commission Justice (2013). '*Protection of personal data*' *[*Online] Available at: http://ec.europa.eu/justice/data-protection/ [Accessed 18 September 2013].

Foster, I.; Yong Zhao; Raicu, I.; Shiyong Lu., (2008). '*Cloud Computing and Grid Computing 360-Degree Compared*', Grid Computing Environments Workshop, 2008. GCE '08 , vol., no., pp.1,10, 12-16 Nov. 2008

Gartner Inc., (2013). '*Forecast Overview: Public Cloud Services, Worldwide, 2011-2016, 2Q12 Update*' [pdf] Available at: http://www.cloudpeople.it/wp-content/uploads/forecast_overview_public_clo_234817.pdf. [Accessed 20 September 2013].

Google. (2013). '*Google Analytics*'. [ONLINE] Available at: http://www.google.com.mt/analytics/. [Accessed 04 November 13].

Graham, G., (2011). '*Promises and illusions of data protection in India. International Data Privacy Law*', vol.1(1), pp. 47-69.

Guha R. & Al-Dabass D., (2010). '*Impact of Web 2.0 and Cloud Computing Platform on Software Engineering*'. 2010 International Symposium on Electronic System Design.

Gunther N.J.(1993) '*Guerrilla Capacity Planning: A Tactical Approach to Planning for Highly Scalable Applications and Services*.' Springer, Heidelberg, Germany, 1st edition, 2007.

IEEE Standard for a Software Quality Metrics Methodology, (1998), '*IEEE Std 1061-1998*' , vol., no., pp.i, 31 Dec. 1998

International Data Center, (2010). '*Cloud Computing to Drive $6.4 Billion in Server Hardware Spending by 2014*'. [press release] July 30, 2010. Available at: http://www.idc.com/about/viewpressrelease.jsp?containerId=prUS22440510 [Accessed 10/10/2013]:

International Organization for Standarization (2001),'*ISO Standard 9126: Software Engineering – Product Quality, parts 1, 2 and 3*', International Organization for Standarization, Geneve, 2001 (part 1), 2003 (parts 2 and 3).

Jackson K.R. , Ramakrishnan L., Muriki K., Canon S., Cholia S., Shalf J. , Wasserman H.J., and Wright N.J., (2010). '*Performance analysis of high performance computing applications on the AmazonWeb services Cloud*,' Proc. CloudCom 2010, IEEE Computer Society, Nov.-Dec. 2010, pp. 159–168.

Kavalionak, H., & Montresor A (2012). '*P2P and cloud: A marriage of convenience for replica management*'. Self-Organizing Systems. Springer Berlin Heidelberg, 2012. 60-71.

Khalil, Issa M.; Khreishah, Abdallah; Bouktif, Salah; Ahmad, Azeem, "Security Concerns in Cloud Computing," Information Technology: New Generations (ITNG), 2013 Tenth International Conference on , vol., no., pp.411,416, 15-17 April 2013

Kherajani M. & Gupta H. & Shrivastava A., 2012. An efficient Cost Calculation Mechanism for Cloud and non-cloud Computing environment in Java. *2012 International conference on computer technology and Science,* vol. 47.

Koomey J, 2011. *Growth in Data center electricity use 2005 to 2010* [Online]. Available at: http://www.analyticspress.com/datacenters.html [Accessed 17th August 2013]

Kotsis, G. (2006). *Performance of Web Applications*. In Kappel, G., Proll, B., Reich, S. and Retschitzegger, W. (eds.) Web Engineering. John Wiley & Sons Ltd., Chicester (2006)

Lanois, P. (2010). *Caught in the clouds: The Web 2.0, cloud computing, and privacy*. Nw. J. Tech. & Intell. Prop., 9, 29.

Lee Y. J. & Lee Y. J. & Cheun W. D. & Kim D. S., 2009, A quality model for Evaluating Software-as-a-service in cloud computing, *Seventh ACIS International Conference on Software Engineering Research, Management and Applications*

Li A., Yang X., Kandula S., Zhang M. (2011). *Comparing Public-Cloud Providers*, Internet Computing, IEEE , vol.15, no.2, pp.50,53, March-April 2011

Miller, R. B. (1968). *Response time in man-computer conversational transactions*. Proc. AFIPS Fall Joint Computer Conference Vol. 33, 267-277.

Miller, R., 2009. Pew: 69 Percent of Americans Use Cloud Apps. [Online] Available at: http://www.datacenterknowledge.com/archives/2008/09/12/pew-data-shows-solid-uptake-for-cloud-apps/ [Accessed 29 09 2013].

Nielsen J. (2013). *Response Time Limits* [ONLINE] Available at: http://www.nngroup.com/articles/response-times-3-important-limits/. [Accessed 21 October 2013].

Oracle, 2005. *Scalability and High Availability* [Online]. Available at: http://docs.oracle.com/cd/B14099_19/core.1012/b13994/avscalperf.htm [Accessed 17th August 2013]

Oracle., 2012. *Ten Questions to Ask Your Cloud Vendor Before Entering the Cloud* . [ONLINE] Available at: http://www.oracle.com/us/products/applications/10-questions-for-cloud-vendors-1639601.pdf. [Accessed 18 September 13].

Oxford Economics, 2013. *Economic Potential of Developing Cloud Computing Industry in Northern Ireland* [pdf] Available at: http://www.investni.com/cloud_computing_economic_potential_report.pdf. [Accessed 20 September 2013].

Priya, B.; Pilli, E.S.; Joshi, R.C.(2013). A survey on energy and power consumption models for Greener Cloud, *Advance Computing Conference (IACC), 2013 IEEE 3rd International* , vol., no., pp.76,82, 22-23 Feb. 2013

Qian H, Medhi D. & Trivedi K (2011). A Hierarchical Model to Evaluate Quality of Experience of Online Services hosted by Cloud Computing, *IEEE International Symposium on Integrated Management, 2011 IFIP/IEEE International Symposium on* , Vol., No., pp.105,112, 23-27 May 2011

Roloff, E.; Diener, M.; Carissimi, A.; Navaux, P.O.A., (2012). *High Performance Computing in the cloud: Deployment, performance and cost efficiency*, Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on , vol., no., pp.371,378, 3-6 Dec. 2012

Sabahi, F. (2011). Cloud computing security threats and responses, *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on* , vol., no., pp.245,249, 27-29 May 2011

Salesforce.com.(N/A). *Agile Development Meets Cloud Computing for Extraordinary Results*.[pdf]. Available at: http://www.developerforce.com/media/ForcedotcomBookLibrary/WP_Agile_112608.pdf [Accessed 22 August 2013]

SandHill Group (2012). *Sand Hill Group Study Finds Massive Job Creation Potential through Cloud Computing | Sandhill*. [ONLINE] Available at: http://sandhill.com/article/sand-hill-group-study-finds-massive-job-creation-potential-through-cloud-computing/ . [Accessed 15 September 2013].

Thorhauge, S., 2012. *How People Spend Their Time Online*. [Online] Available at: http://www.mindjumpers.com/blog/2012/05/time-spend-online/ [Accessed 07 09 2013].

Undheim A.; Chilwan A.; Heegaard P. (2011). *Differentiated Availability in Cloud Computing SLAs*, Grid Computing (GRID), 2011 12th IEEE/ACM International Conference on , vol., no., pp.129,136, 21-23 Sept. 2011

Wang J.; Varman, P.; Changsheng X (2010), *Avoiding performance fluctuation in cloud storage*, High Performance Computing (HiPC), 2010 International Conference on , vol., no., pp.1,9, 19-22 Dec. 2010

Wassan, M; Fayoumi A. (2012). Service Availability Evaluation for a Protection Model in Hybrid Cloud Computing Architecture. *1st IEEE International Symposium on Telecommunication Technologies*, Vol., No., pp., 307-312.

Wu, J.; Liang, L., Qianhui (Althea); and Bertino, E.. (2009). Improving Scalability of Software Cloud for Composite Web Services. *2009 IEEE International Conference on Cloud Computing*. . Research Collection School Of Information Systems.

Xiaowei Yang A. L., Ming Zhang S.K., (2010), *Comparing Public Cloud Providers,* Internet Computing, IEEE , vol.15, no.2, pp.50,53, March-April 2011

Yau S. & An G, 2011. Software engineering meets services and Cloud Computing, *IEEE Computer Society,* Vol. 44(10), pp. 47-53, 2011.

Zia A. & Ahmad-Khan M. N. (2012) *Identifying Key Challenges in Performance Issues in Cloud Computing,* I.J. Modern Education and Computer Science, vol. 10, pp. 59-68, Sept. 2012.

Zhang S.; Chen X.; Zhang S.; Huo X. (2010), *The comparison between cloud computing and grid computing,* Computer Application and System Modeling (ICCASM), 2010 International Conference on , vol.11, no., pp.V11-72,V11-75, 22-24 Oct. 2010

Zheng Li; O'Brien, L.; He Zhang; Cai, R.(2013). Boosting Metrics for Cloud Services Evaluation -- The Last Mile of Using Benchmark Suites, *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference*, vol., no., pp.381,388, 25-28 March 2013

BIBLIOGRAPHY:

Adekunle, Y.A, Maitanmi, S.O, Malasowe, B.O, Osundina, S. A, 2012. *Economics of Cloud Computing*. International Journal of Engineering and Innovative Technology, Volume 1, Issue 4, April 2012, 4-7.

AirBrake Inc.. 2013. AirBrake. [ONLINE] Available at: https://www.airbrake.io/about. [Accessed 04 November 13].

Alertra. 2013. Website Monitoring Service. [ONLINE] Available at: http://www.alertra.com/. [Accessed 04 November 13].

Amazon Web Services, 2013 . *Amazon EC2 Pricing*. [Online] Available at: http://aws.amazon.com/ec2/pricing/ [Accessed 02 09 2013].

Amazon Web Services, 2013. *AWS Elastic Beanstalk Pricing* [Online] Available at: http://aws.amazon.com/pricing/elasticbeanstalk/ [Accessed 14 May 2013]

Anon, n.d. *ISO 9126 Software Quality Characteristics* [Online] Available at: http://www.sqa.net/iso9126.html [Accessed 15 May 2013]

AppFail. 2013. Rich failure analytics in our web dashboard. [ONLINE] Available at: http://appfail.net/. [Accessed 04 November 13].

AppHarbor, n.d. *Pricing* [Online] Available at: https://appharbor.com/pricing [Accessed 14 May 2013]

Apple Inc.. 2013. iCloud. [ONLINE] Available at: http://www.apple.com/icloud/ . [Accessed 04 November 13].

Apple Inc.. 2013. iTunes. [ONLINE] Available at: http://www.apple.com/itunes/. [Accessed 04 November 13].

Bartels, A., 2012. Rackspace Cloud Sites – A Look at the Numbers. [Online] Available at: http://www.rackspace.com/blog/rackspace-cloud-sites-a-look-at-the-numbers/ [Accessed 01 10 2013].

Commision, E., 2013. Protection of personal data. [Online] Available at: http://ec.europa.eu/justice/data-protection/ [Accessed 18 08 2013].

Derby University, n.d. *Call for Chapters - Software Engineering Frameworks for Cloud Computing Paradigm* [Online] Available at: http://www.derby.ac.uk/computing/research/software-engineering-frameworks-for-cloud-computing-paradigm [Accessed 15 May 2013]

ESRI, 2012. *ArcGIS Enterprise Systems: Performance and Scalability* [pdf] Available at: http://proceedings.esri.com/library/userconf/eurouc12/papers/euc_28.pdf [Accessed 10 August 2013]

Fazal-e-Amin ; Mahmood, A.K.; Oxley, A. (2012), "An evolutionary study of reusability in Open Source Software," *Computer & Information Science (ICCIS), 2012 International Conference on*, vol.2, no., pp.967,972, 12-14 June 2012

Fershtman C., Gandal N., 2012. *Migration to the cloud ecosystem: Ushering in a new generation of platform competition* [Online]. Available at http://www.voxeu.org/article/cloud-computing-economic-issues [Accessed 17th August 2013]

GoDaddy. 2013. *Dedicated servers.* [ONLINE] Available at: http://www.godaddy.com/hosting/dedicated-servers.aspx?isc=gtnieu26&ci=9014. [Accessed 27 September 13]

GoGrid, 2013. *Pricing.* [Online] Available at: http://www.gogrid.com/products/pricing [Accessed 20 10 2013].

Golden B., 2013. *What Economists Can Teach Us About Cloud Computing* [Online]. Available at http://www.cio.com/article/735995/What_Economists_Can_Teach_Us_About_Cloud_Computing [Accessed 17th August 2013]

Google, N/A. *Google Cloud Storage Pricing.* [Online] Available at: https://cloud.google.com/pricing/cloud-storage [Accessed 15 10 2013].

Haff G., 2008. What is Cloud Computing?. *The difference between Web Hosting and Cloud Computing* [blog] 24 April. Available at: http://www.qrimp.com/blog/blog.The-difference-between-Web-Hosting-and-Cloud-Computing.html [Accessed 15 May 2013]

Hat, R., 2013. *Are there different gear sizes and how much do they cost?.* [Online] Available at: https://www.openshift.com/faq/are-there-different-gear-sizes-and-

how-much-do-they-cost
[Accessed 20 10 2013].

Heriot Watt University, 2008. A Guide to writing your Masters Dissertation. [pdf] Available at: http://www.sml.hw.ac.uk/postgraduate/downloads/dissertations/dissertationguide.pdf [Accessed 01 October 2013].

Heroku, N/A. *Pricing*. [Online] Available at: https://www.heroku.com/pricing [Accessed 22 09 2013].

Holt, R., 2013. Twitter in numbers. [Online] Available at: http://www.telegraph.co.uk/technology/twitter/9945505/Twitter-in-numbers.html [Accessed 21 09 2013].

Hong Zhou; Yang, H.; Hugill, A. (2010). *An Ontology-Based Approach to Reengineering Enterprise Software for Cloud Computing*, *Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual* , vol., no., pp.383,388, 19-23 July 2010

Hosted Graphite, 2012. *Hosted Graphite Documentation* [Online] Available at: http://docs.hostedgraphite.com/ [Accessed 07 July 2013]

HP, 2013. *HP Cloud Pricing*. [Online] Available at: https://www.hpcloud.com/pricing [Accessed 08 10 2013].

Huei Liew S. & Su Y., 2012. CloudGuide: Helping Users Estiamte Cloud Deployment Cost and Performance for Legacy Web Applications. *2012 International Conference on Cloud Computing Technology and Science.*

IPage. 2013. *Dedicated Server Pricing*. [ONLINE] Available at: http://www.ipage.com/knowledgebase/beta/article.bml?ArticleID=1913. [Accessed 26 September 13].

JustHost. 2013. *Dedicated hosting prices*. [ONLINE] Available at: https://my.justhost.com/cgi/help/pricing-dedicated. [Accessed 27 September 13].

Lee Y. J.; Kim S. D. (2010), "Software Approaches to Assuring High Scalability in Cloud Computing," e-*Business Engineering (ICEBE), 2010 IEEE 7th International Conference on* , vol., no., pp.300,306, 10-12 Nov. 2010

Liew S. H. & Su Y. 2012. CloudGuide: Helping Users Estimate Cloud Deployment Cost and Performance for LegacyWeb Applications, *IEEE 4th International Conference on Cloud Computing Technology and Science*

Matzefriedrich, 2013. *CODE METRICS VIEWER 2 CTP*. [Online] Available at: http://codemetricsviewer.wordpress.com/tag/code-metrics/ [Accessed 20 09 2013].

Maynard Claire, (2012), *Scrum & Agile Basics* [ONLINE]. Available at: https://www.cprime.com/resources/what-is-agile-what-is-scrum/ [Accessed 09 September 13].

Microsoft, 2013. *Code Metrics Viewer 2010* [Online]. Available at http://visualstudiogallery.msdn.microsoft.com/9f35524b-a784-4dbc-bd7b-6babd7a5a3b3 [Accessed 17th August 2013]

Microsoft, 2013. Pricing. [Online] Available at: http://www.windowsazure.com/en-us/pricing/calculator/ [Accessed 15 10 2013].

Ookla (2013). *Speedtest.net by Ookla - The Global Broadband Speed Test*. [ONLINE] Available at: http://www.speedtest.net/ . [Accessed 25 November 2013].

Percona, 2010. *Forecasting MySQL Scalability with the Universal Scalability Law*.[pdf] Available at: http://www.percona.com/files/white-papers/forecasting-mysql-scalability.pdf [Accessed on 15th August 2013]

Politis, D. 2013, BetterCloud Blog. 13 November 2012. *Google Apps: The Numbers You Need to Know: Blog*. Available from: http://blog.bettercloud.com/google-apps-survey/ [29 October 2013].

Press, T. A., 2013. Number of active users at Facebook over the years. [Online] Available at: http://news.yahoo.com/number-active-users-facebook-over-230449748.html [Accessed 03 09 2013].

RackSpace, N/A. *Pricing*. [Online] Available at: http://www.rackspace.com/cloud/servers/#pricing [Accessed 20 10 2013].

Retschitzegger, W. (Eds.). *Web Engineering*, pp. 247-264. Heidelberg: John Wiley & Sons.

Ryan K.L. Ko, Stephen S.G. Lee, Veerappa Rajan, 2012. *Understanding Cloud Failures* [Online] Available at: http://spectrum.ieee.org/computing/networks/understanding-cloud-failures [Accessed 22 09 2013].

Seventh Framework Programme, 2011. Cloud computing, Internet of Services and Advanced Software Engineering, *European Commission Information Society and Media*.

Shilov, A., 2010. *Cloud Computing to Drive $6.4 Billion in Server Hardware Spending by 2014 - Analyst.*. [Online] Available at: http://www.xbitlabs.com/news/other/display/20100809210619_Cloud_Computing_to_Drive_6_4_Billion_in_Server_Hardware_Spending_by_2014_Analyst.html [Accessed 21 09 2013].

Smart Cloud, 2011. *IBM SmartCloud Application Services – FAQ*. [pdf] Available at: http://www-05.ibm.com/it/cloud/assets/IBMSmartCloud_Application_Services_FAQ.pdf [Accessed on 12th October 2013]

Tosic V. & Wada H. & Guabtni A. & Lee K. & Liu A., 2011. Management towards Reducing Cloud Usage Costs

Uhuru Software, 2012. *Ready to Go Pricing* [Online] Available at: http://uhurusoftware.com/ready-to-go-pricing/ [Accessed 14 May 2013]

Vitek J. A., n.d. Service oriented cloud computing Architecutres, *UMM CSci Senior Seminar Conference*.

VMWare, 2013. *VMware vFabric Hyperic Overview*. [pdf] Available at: http://www.vmware.com/pdf/vFabric-Hyperic-Overview-57.pdf [Accessed 10 August 2013]

Wind S. 2011. Open Source Cloud computing Management Platforms: Introduction, Comparision and Recommendations for Implementation, *IEEE Conference on Open Systems*.

Windows Azure, 2013. *Purchase Options* [Online] Available at: http://www.windowsazure.com/en-us/pricing/purchase-options/ [Accessed 14 May 2013]

Yard, E., N/A. *Custom configuration.* [Online] Available at: https://www.engineyard.com/products/cloud/pricing [Accessed 20 10 2013].

Younge, A.J.; von Laszewski, G.; Lizhe Wang; Lopez-Alarcon, S.; Carithers, W. (2010). *Efficient resource management for Cloud computing environments*, Green Computing Conference, 2010 International , vol., no., pp.357,364, 15-18 Aug. 2010

Zheng Li; O'Brien, L.; He Zhang; Cai, R., (2012). *On a Catalogue of Metrics for Evaluating Commercial Cloud Services*, Grid Computing (GRID), 2012 ACM/IEEE 13th International Conference on , vol., no., pp.164,173, 20-23 Sept. 2012